

Fig. 1

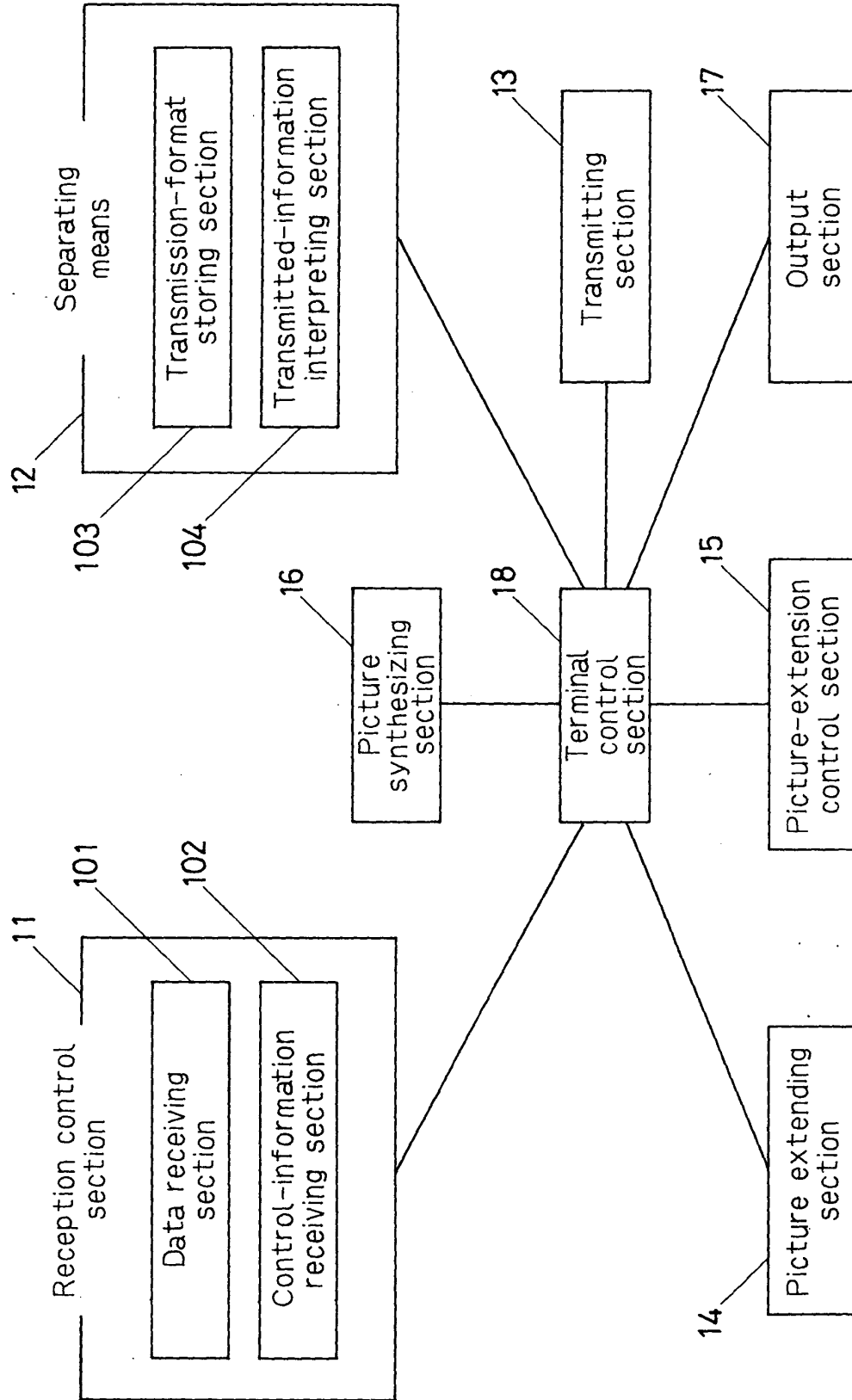
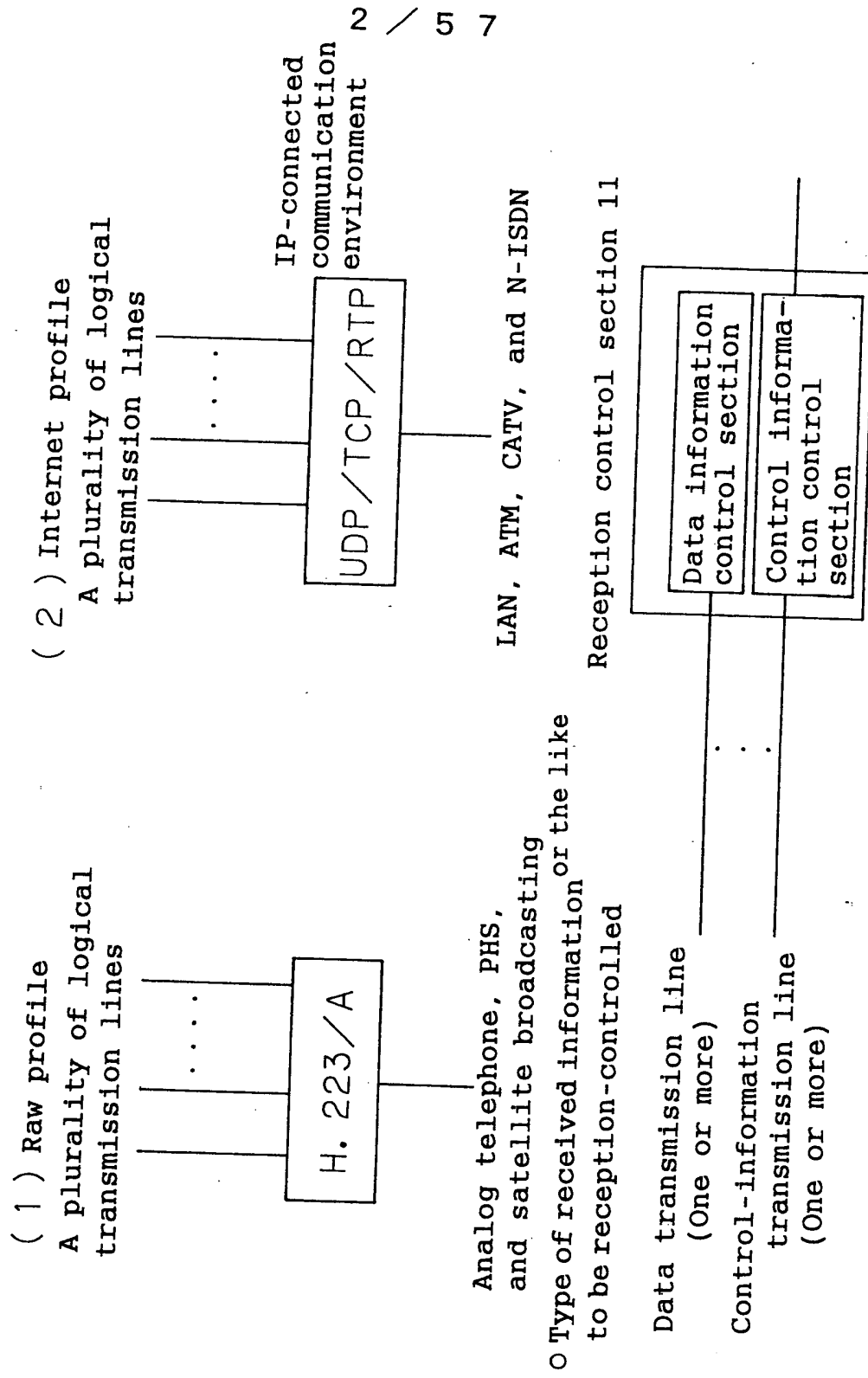


Fig. 2



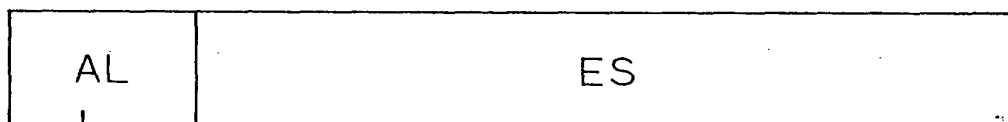

- Information showing start position capable of processing data or not
 - Flag for random access (Random access flag), e.g. Intra-frame (I-picture) in the case of picture
 - Flag showing access unit (Access flag), e.g. Frame in the case of picture, GOB unit

AL : Adaptation layer

ES : Elementary stream

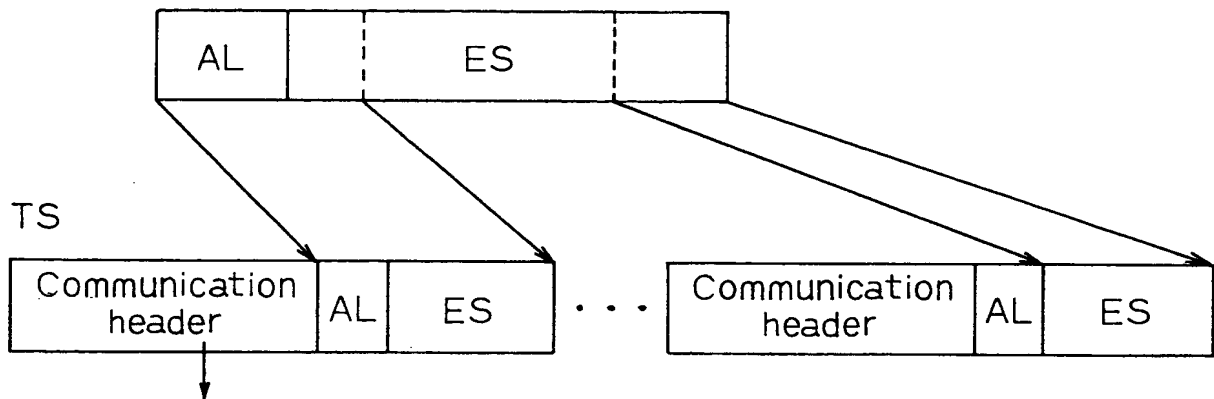
PTS : Presentation time stamp

Header information of data Data (Picture or sound for each frame)



- ↓
- Information showing start position capable of processing data or not
 - Information showing data reproducing time (PTS)
 - Information showing data processing priority

oTS:Transport stream(Transmission packet)



- Information showing start position capable of processing pieces of data or not
- Identification number for showing data sequence(Sequence number)
- Time concerned with transmission of pieces of data

oHandling time stamp and marker bit

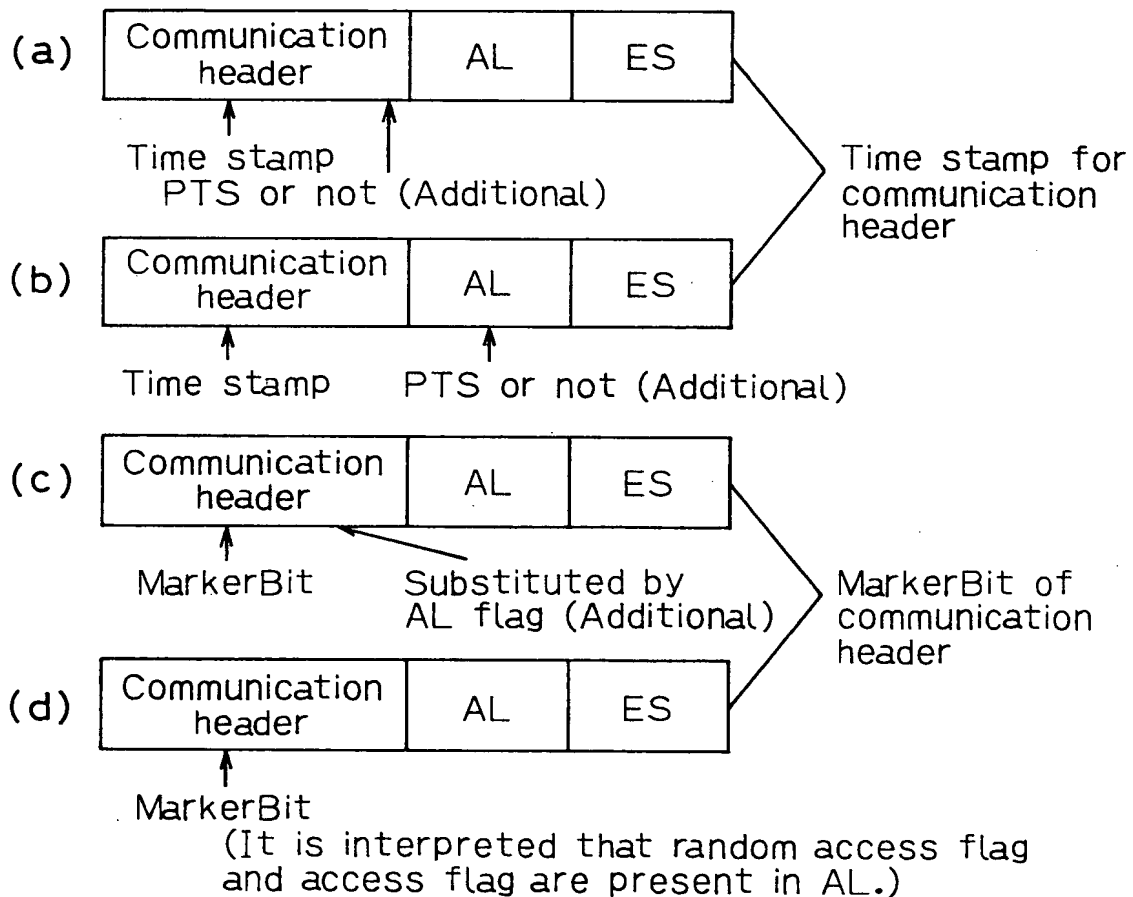


Fig. 5(a) 5 / 57

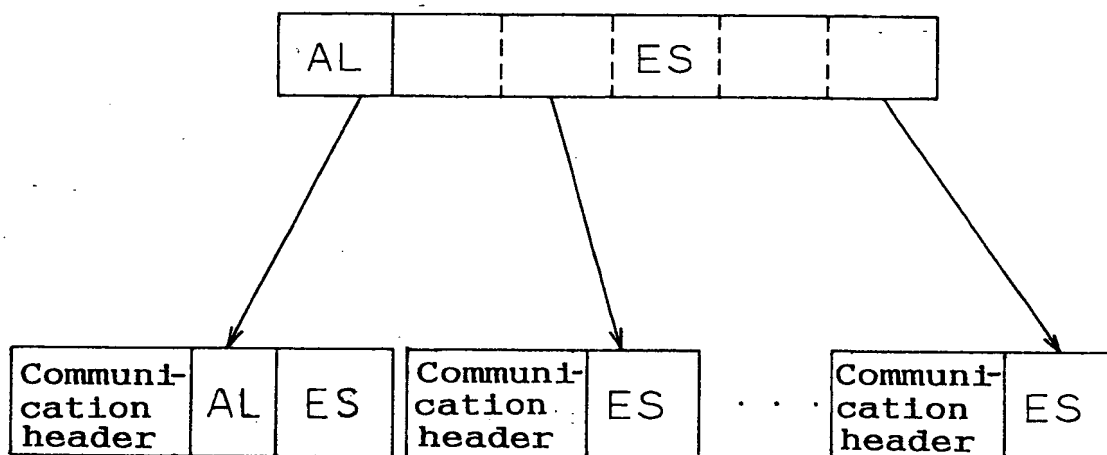


Fig. 5(b)

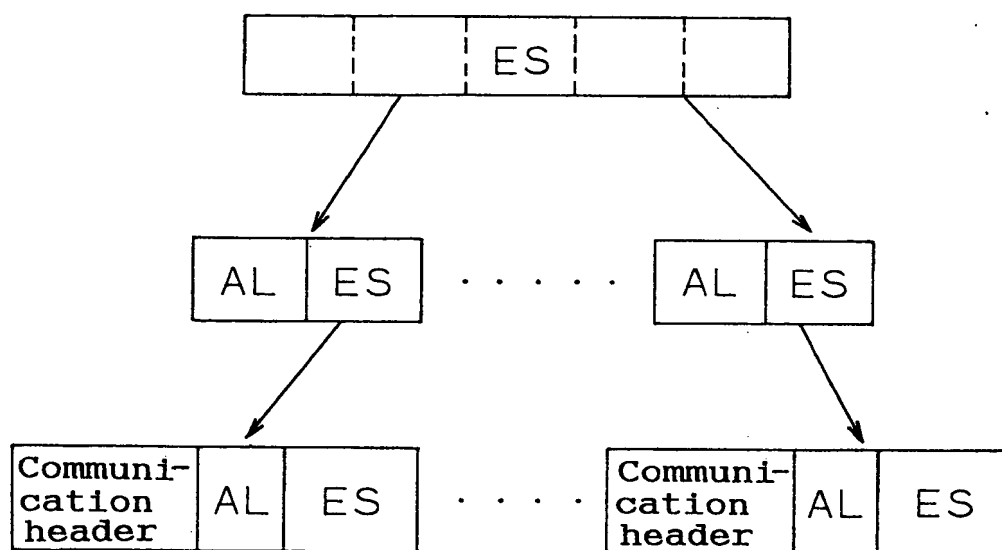


Fig. 6(a)

Method for making the most use of RTP base

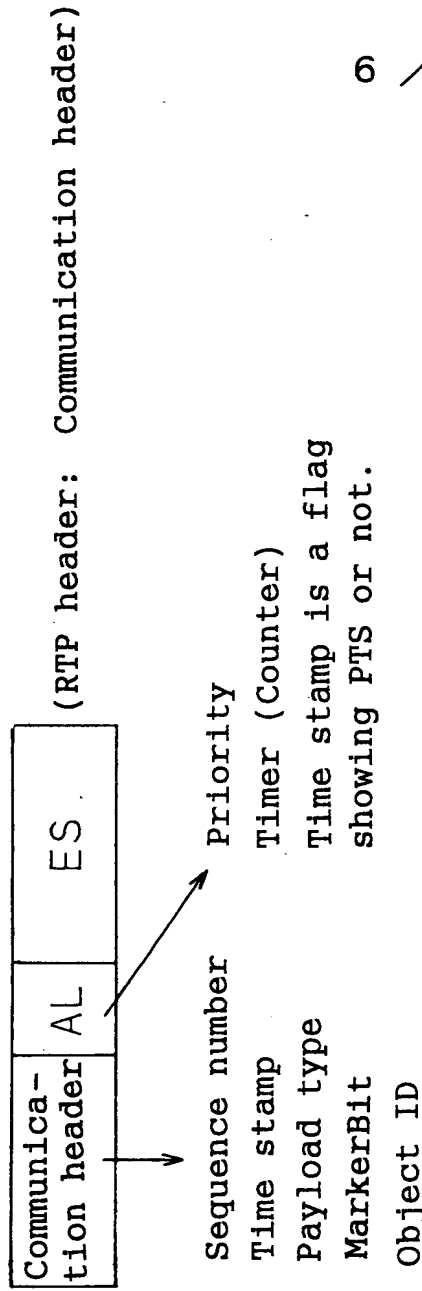


Fig. 6(b)

Method for simplifying the communication header

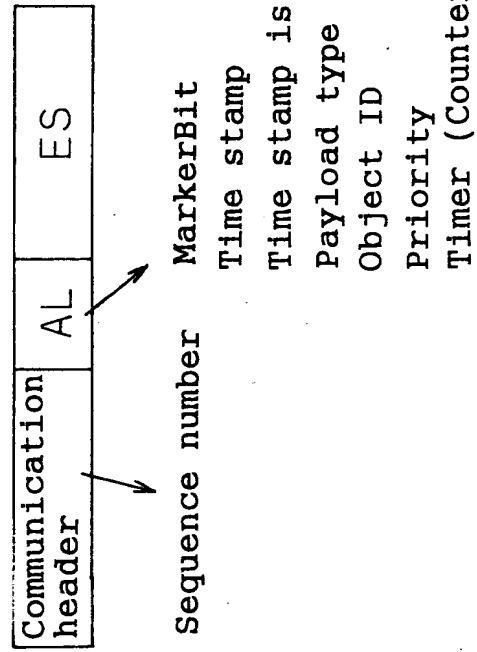


Fig. 6(c)

Method for changing every AL information to communication header

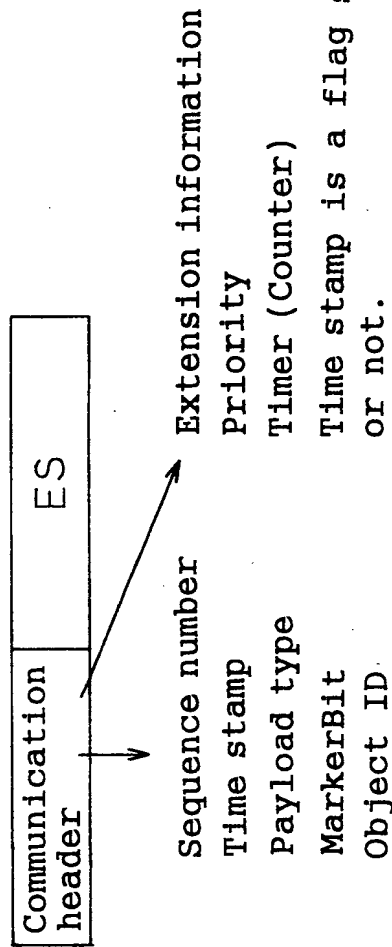
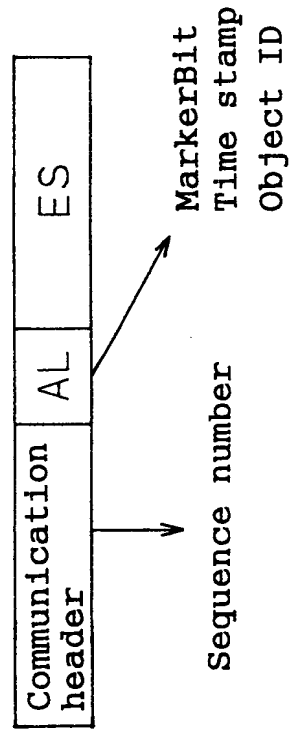
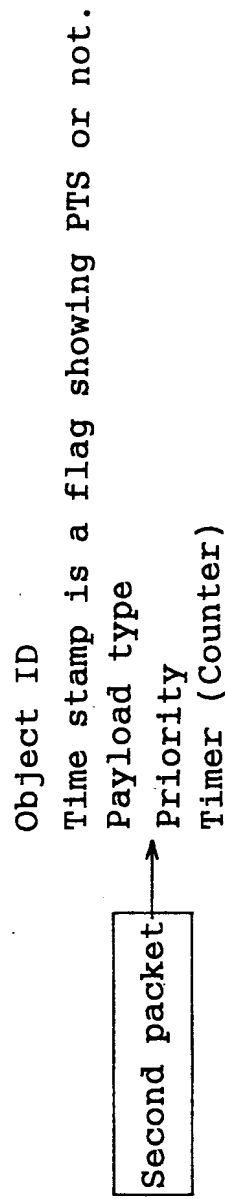


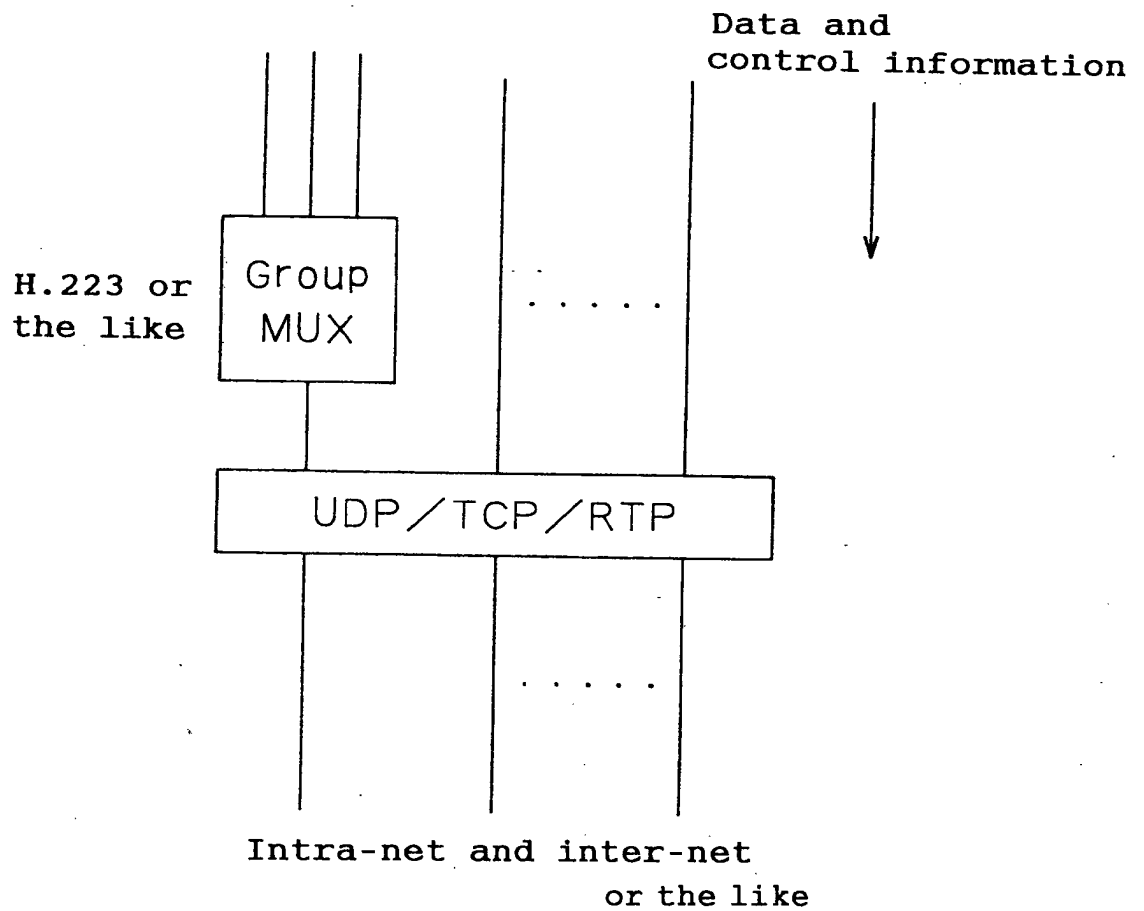
Fig. 6(d)

Method for transmitting as another packet



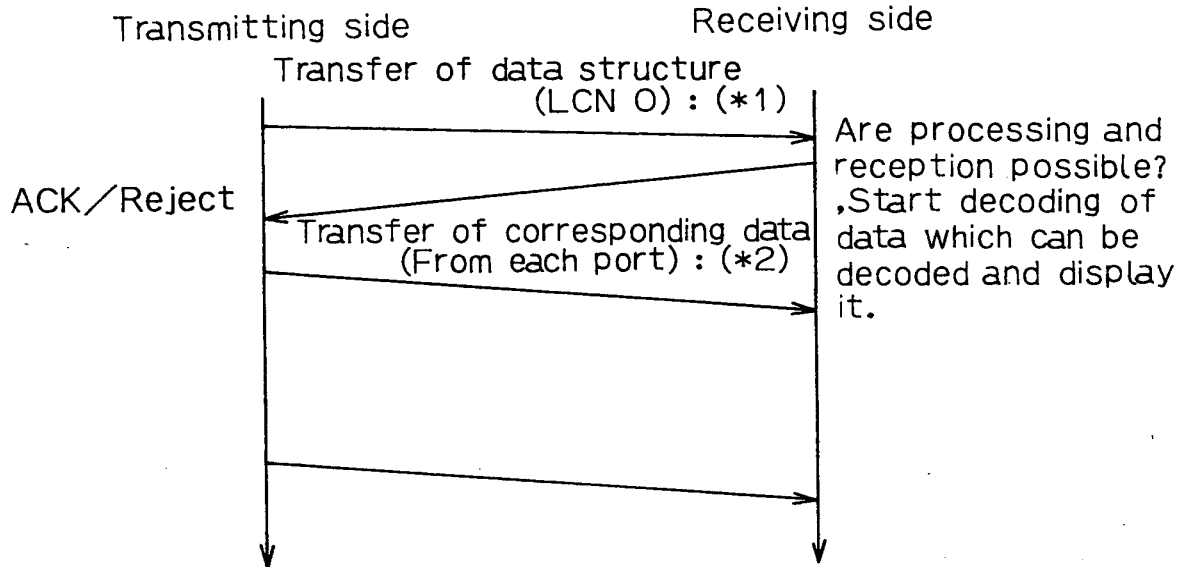
F i g . 7

8 / 5 7

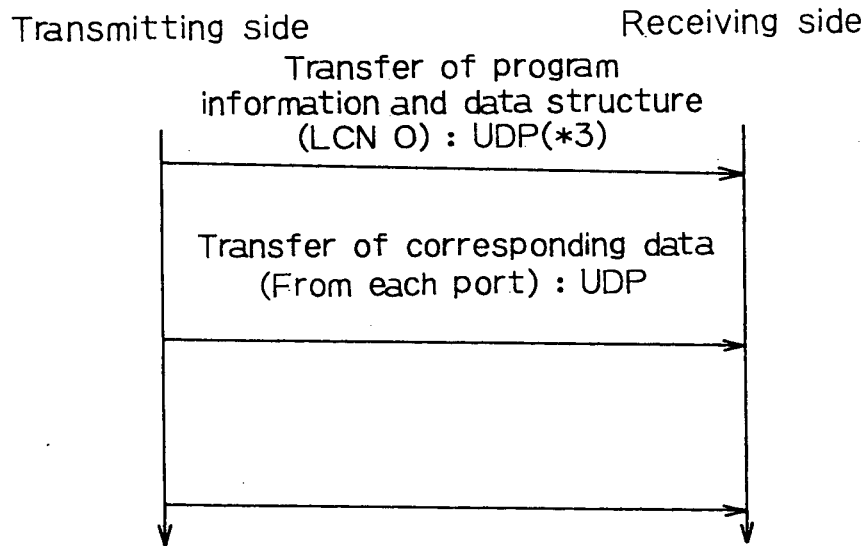


• Broadcast program transmitting procedure

<Broadcast type and communication type including return channel>



<Broadcast type (with no return channel)>



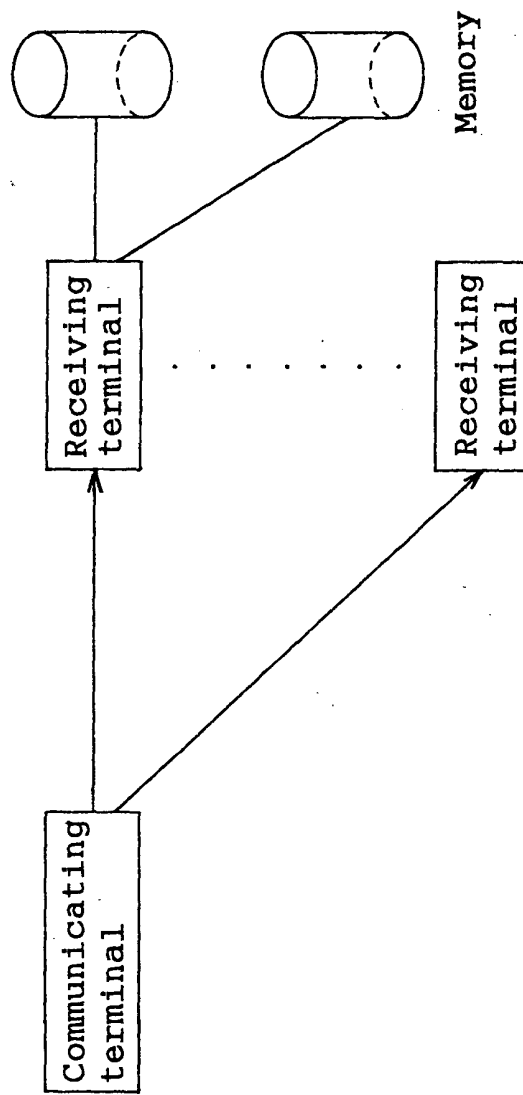
(*1) Must be a system for detecting and retransmitting a packet loss like TCP.

(*2) RTP/RTCP or TCP/IP

(*3) Same data (picture or sound) or control information (broadcast program or data structure) is continuously repeatedly transmitted. A packet is detected and sequence is kept at a receiving terminal in accordance with a sequence number. (To be used in a local closed region. Traffic becomes too large.)

Fig. 9(a)

When program or data is present at a receiving terminal



Program or data

Program or data identifier to be required

Flag, counter, or timer for communicating a point of time to be required

Fig. 9(b)

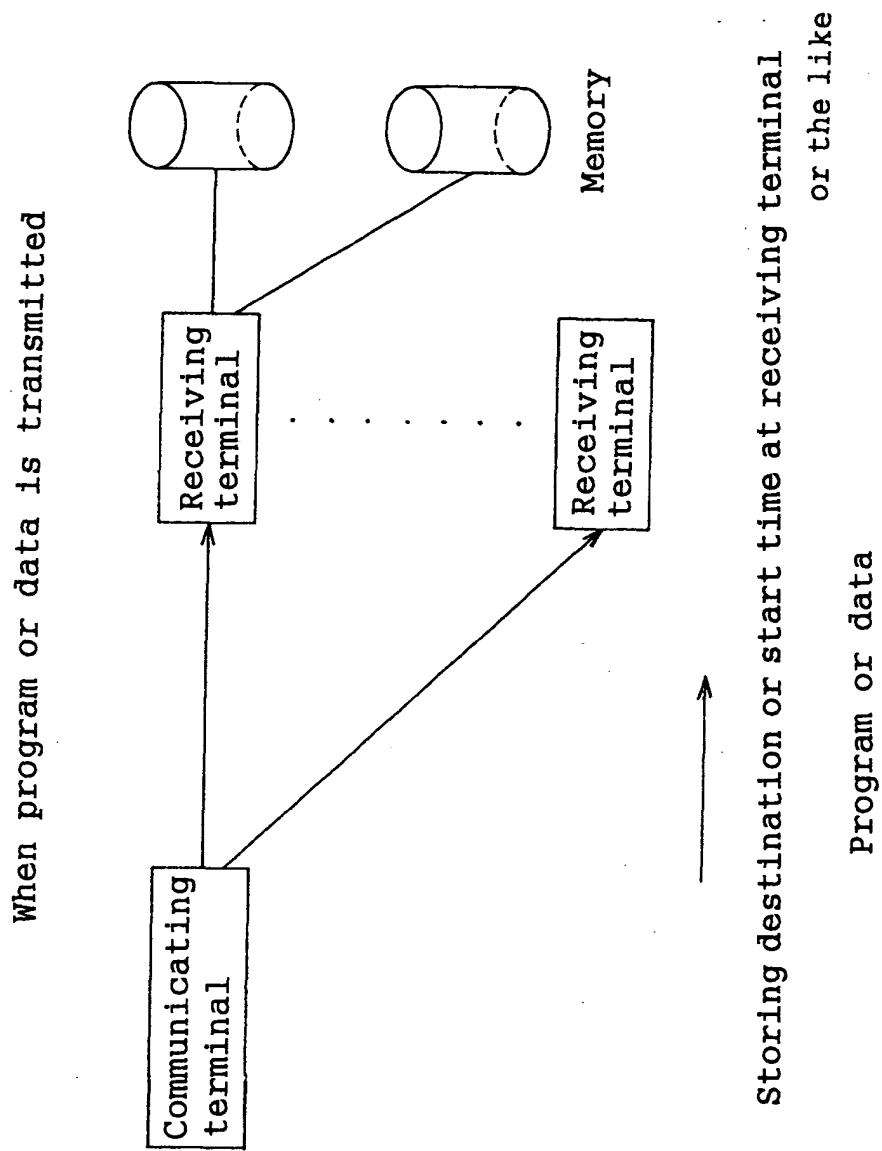


Fig. 10(a)

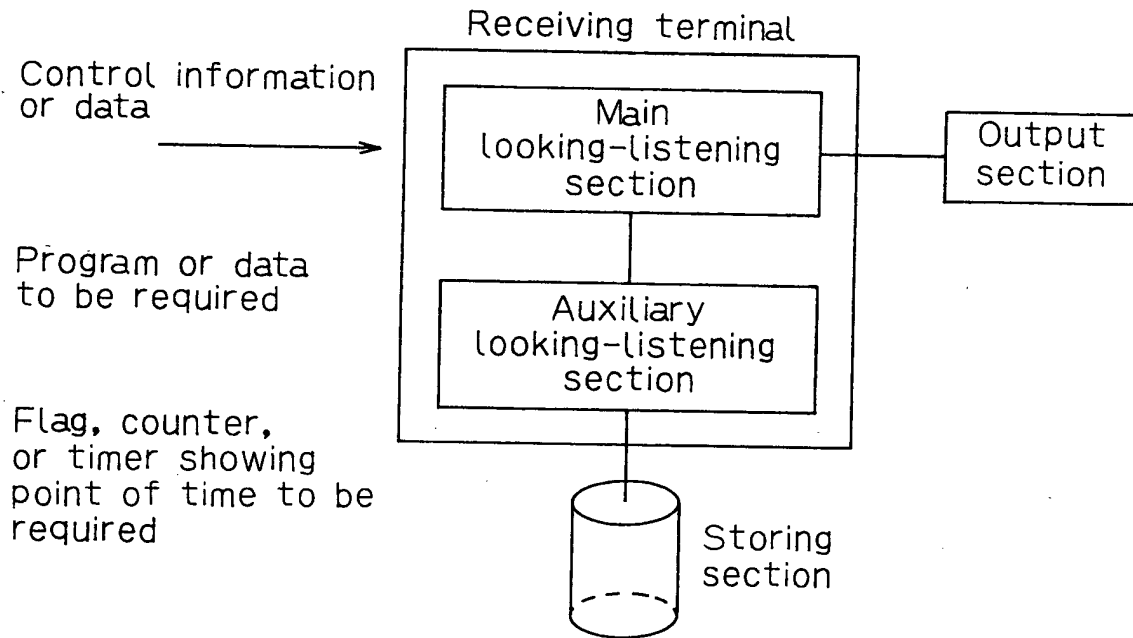
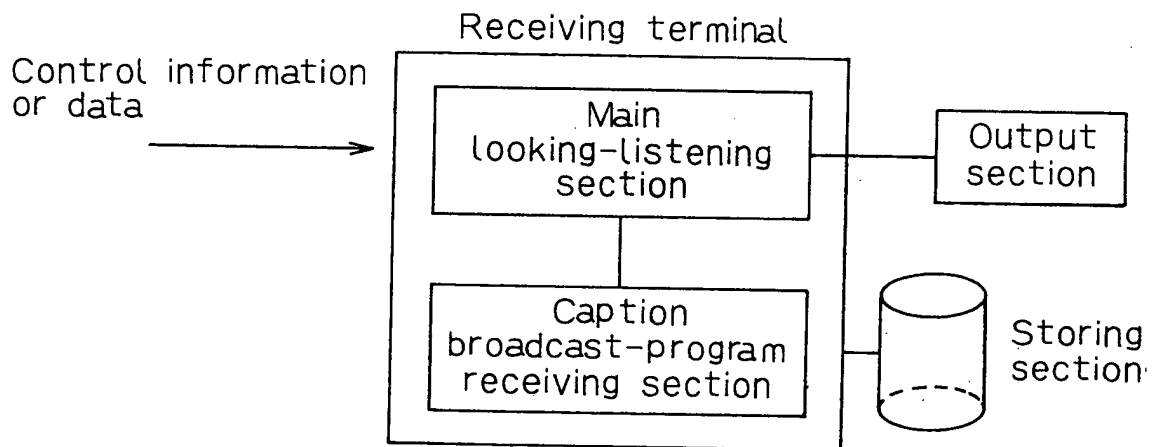
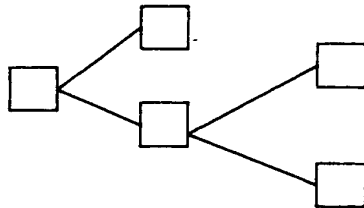


Fig. 10(b)



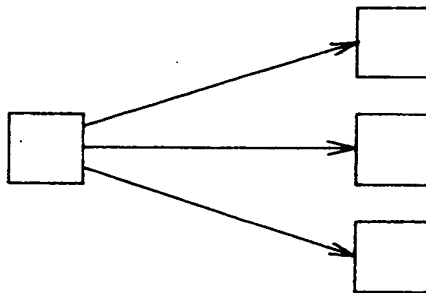
F i g . 1 1 (a) ^{1 3 / 5 7}

<Hierarchical image of object>



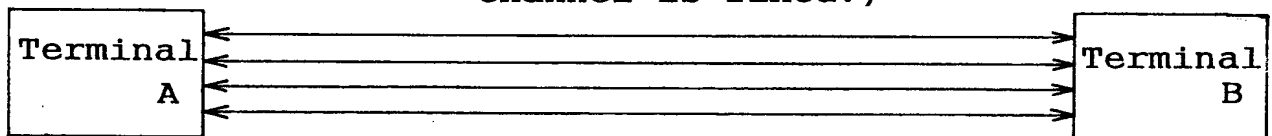
<Transmission image of object>

<1. Broadcast type>



<2. Communication type>

RTP/RTCP (Program ID of each logical channel is fixed.)



LCNO (control)

F i g . 1 1 (b)_{1 4 / 5 7}

```

=====
- Capability exchange definitions (original from H.245)
=====
TerminalCapabilitySet ::= SEQUENCE
{
    sequenceNumber          SequenceNumber,

    multiplexCapability      MultiplexCapabilityOPTIONAL
    capabilityTable          SET SIZE(1..256) OF Capability
                             TableEntryOPTIONAL,
    capabilityDescriptors    SET SIZE(1..256) OF Capability
                             DescriptorOPTIONAL,
    mpeg4Capability          MPEG4CapabilityOPTIONAL.
    ...
}

```

```

=====
=
-MPEG4 Capability definitions
=====
=
MPEG4Capability ::=SEQUENCE
{
    sequenceNumber          SequenceNumber,
    NumberOfProcessObject   SEQUENCE
    {
        MaxNumberOfVideo    INTEGER(0..1023),
        ...
        MaxNumberOfSounds    INTEGER(0..1023),
        ...
        MaxNumberOfMux        INTEGER(0..1023),
    }
    reconfigurationALCapability  BOOLEAN,
    ...
}
MPEG4CapabilityAck ::=SEQUENCE
{
    sequenceNumber          SequenceNumber,
    ...
}
MPEG4CapabilityReject ::=SEQUENCE
{
    sequenceNumber          SequenceNumber,
    NumberOfProcessObject   SEQUENCE
    {
        maxNumberOfVideo    MaxNumberOfVideo,
        ...
        maxNumberOfSounds    MaxNumberOfSounds
        ...
        MaxNumberOfMux        maxNumberOfMux,
    }
    reconfigurationALCapability  BOOLEAN,
    ....
}

```

1 6 / 5 7 **F i g . 1 3 (a)**

```

=====
=
-Group MUX definitions
=====
=
CreateGroupMux                                ::=SEQUENCE
{
    sequenceNumber                            SequenceNumber,
    GroupMuxID                                INTEGER(0..1023),
    lanportNumber                             LANPortNumber,
    ...
}
CreateGroupMuxAck                             ::=SEQUENCE
{
    sequenceNumber                            SequenceNumber,
    ...
}
CreateGroupMuxReject                         ::=SEQUENCE
{
    sequenceNumber                            SequenceNumber,
    cause                                     CHOICE
    {
        ...
    }
    ...
}

```


F i g . 1 3 (b)^{1 7 / 5 7}

DestoryGroupMux	::=SEQUENCE
{	
sequenceNumber	SequenceNumber,
GroupMuxID	INTEGER(0..1023),
...	
}	
DestoryGroupMuxAck	::=SEQUENCE
{	
sequenceNumber	SequenceNumber,
...	
}	
DestoryGroupMuxReject	::=SEQUENCE
{	
sequenceNumber	SequenceNumber,
cause	CHOICE
{	
...	
}	
...	
}	

F i g . 1 3 (c)

```

PortNumberStructure ::=SEQUENCE
{
    sequenceNumber          SequenceNumber,
    lanPortNumber           LANPortNumber,
    numberOfLogicalNumber   INTEGER(1..15),
    SEQUENCE SIZE(1..15) OF PortStructureElement,
    ...
}
PortStructureElement ::=SEQUENCE
{
    logicalPortNumber       LogicalPortNumber,
    ...
}
PortNumberStructureAck ::=SEQUENCE
{
    sequenceNumber          SequenceNumber,
    ...
}
PortNumberStructureReject ::=SEQUENCE
{
    sequenceNumber          SequenceNumber,
    cause                   CHOICE
    {
        ...
    }
    ...
}

```

Fig. 14 19 / 57

```

=====
=
  -Logical channel signalling definitions(original from H.245)
  -MPEG4 Object Create Operation(for LANPortNumber)
=====
=
OpenLogicalChannel ::=SEQUENCE
{
  fowardLogicalChannelNumber LogicalChannelNumber,
  fowardLogicalChannelParameters SEQUENCE
  {
    portNumber INTEGER(0..65535)OPTIONAL,
    dataType DataType,
    multiplexParameters CHOICE
    {
      h222LogicalChannelParameters H222LogicalChannelParameters,
      h223LogicalChannelParameters H223LogicalChannelParameters,
      v76LogicalChannelParameters v76LogicalChannelParameters,
      ...,
      h2250LogicalChannelParameters H2250LogicalChannelParameters,
      h223AnnexALogicalChannelParameters
      H223AnnexALogicalChannelParameters
      MPEG4LogicalChannelParameters MPEG4LogicalChanelParameters,
      ...
    },
    ...,
  },
  ...,
}

```

```
MPEG4LogicalChannelParameters ::=SEQUENCE
{
    -H.225BASE                INTEGER(0..65535),
    LANportNumber              INTEGER(0..255),
    ProgramID                  OCTETSTRING(SIZE(128)),
    ...
}

BroadcastChannelProgram ::=SEQUENCE
{
    sequenceNumber              SequenceNumber,
    numberOfChannelNumber       INTEGER(0..1023),
    SEQUENCE SIZE(1..1023) OF MPEG4LogicalChannelParameters
}

ChangeLogicalChannelAttribute ::=SEQUENCE
{
    sequenceNumber              SequenceNumber
    lanportNumber               LANPortNumber,
    ProgramID                   INTEGER(0..255),
    ...
}

ChangeLogicalChannelAttributeAck ::=SEQUENCE
{
    sequenceNumber              SequenceNumber,
    ...
}

ChangeLogicalChannelAttributeReject ::=SEQUENCE
{
    sequenceNumber              SequenceNumber,
    cause                       CHOICE
    {
        ...
    }
    ...
}
```

F i g . 1 6 (a)

```

=====
=
-MPEG4 Object Class definition
=====
MPEG4 Object Class definition      ::=SEQUENCE
{
    sequenceNumber                  SequenceNumber,
    ProgramID                       INTEGER(0..255),
    NumberOfObjectsList             INTEGER(0..1023),
    SEQUENCE SIZE(1..1023) OF ObjectStructureElement
}
ObjectStructureElement             ::=SEQUENCE
{
    SSRC                            INTEGER(0..16777215),
    LANPortNumber                   INTEGER(1024..5000),
    O                               --forRPT(Video&Sound)
    ScrambleFlag                    BOOLEAN,
    CGDOffset                       INTEGER(0..255),
    MediaType                       INTEGER(0..255),
    ...
}

MPEG4 Object Class definitionAck    ::=SEQUENCE
{
    sequenceNumber                  SequenceNumber,
    ...
}

MPEG4 Object Class definitionReject ::=SEQUENCE
{
    sequenceNumber                  SequenceNumber,
    cause                           CHOICE
    {
        ...
    }
    ...
}

```

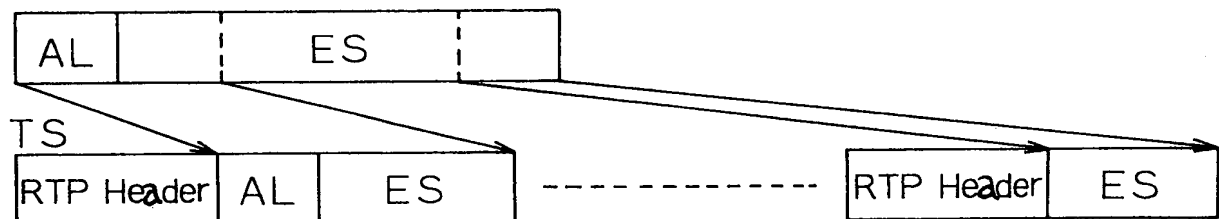
Fig. 16 ^{2 2 / 5 7} (b)

```

=====
=
-Adaptation Layer Reconfiguration Request definitions
=====
ALReconfiguration ::=CHOICE
{
    sequenceNumber          SequenceNumber,
    RandomAccessFlagMaxBit  INTEGER(0...2),
    PresentationTimeStampsMaxBit  INTEGER(0...32),
    CGDPriorityMaxBit       INTEGER(0...8),
                           --for Video and Sound
    ...
}
=====
=
-Adaptation Layer Reconfiguration Response definitions
=====
ALReconfigurationAck ::=SEQUENCE
{
    sequenceNumber          SequenceNumber,
    ...
}
ALReconfigurationReject ::=SEQUENCE
{
    sequenceNumber          SequenceNumber,
    cause                   CHOICE
    {
        ...
    }
    ...
}

```

<Relation between AL, ES, and RTP>



2 3 / 5 7
F i g . 1 7

```
=====
=
-Setup Program and Data Request definitions
=====
Setup Request ::=CHOICE
{
    sequenceNumber      SequenceNumber,
    SSRC IMEGER(0..16777215)2^32,
    Logical Channel Number,  INTEGER(1024...5000),
    setupitem            CHOICE
    {
        executeProgramNumber  INTEGER(0...255),
        dataNumber            INTEGER(0...255),
        executeCommandNumber  INTEGER(0...255),
    },
    nofitycounter           CHOICE
    {
        flag                 BOOLEAN
        counter              INTEGER(0...255),
        timer                INTEGER(0...255),
    },
    ...,
}
}
```

F i g . 1 8

=====

=

-control and AL attribute definitions

=====

ControlALdefinition	::=CHOICE
{	
sequenceNumber	SequenceNumber,
AL	CHOICE
{	
RandomAccessFlagUse	BOOLEAN,
PresentationTimeStampUse	BOOLEAN,
CGDPriorityUse	BOOLEAN,
...	
},	

F i g . 1 9 (a)

```

class ES_header {
    uint(4)    headerID;
    uint(24)   bufferSizeES;
    uint(1)    useTimeStamps;
    .....
    .....
    uint(16)   sequenceNumberMaxBit;
    uint(1)    useHeaderExtension;
    if (useHeaderExtension){
        uint(1)    accessUintStartFlag;
        uint(1)    randomAccessPointFlag;
        uint(1)    OCRsetFlag;
        uint(4)    degradationPriorityMaxBit;
    }
    uint(3)    reserved:
}

```

F i g . 1 9 (b)

```

=====
--Adaptation Layer PDU header configuration Request and Command definition
=====
AL configuration ::= SEQUENCE
{
    sequenceNumber          SequenceNumber,
    defaultHeaderConfiguration  BOOLEAN,
    headerID                INTEGER(0..4),
    MPEG4ALPDUHeaderConfig  SEQUENCE
    {
        accessUintStartFlag  BOOLEAN,
        randomAccessPointFlag  BOOLEAN,
        OCRsetFlag            BOOLEAN,
        degradationPriorityMaxBit  INTEGER(0..4),
        ...
    }
}

```

Fig. 20(a)

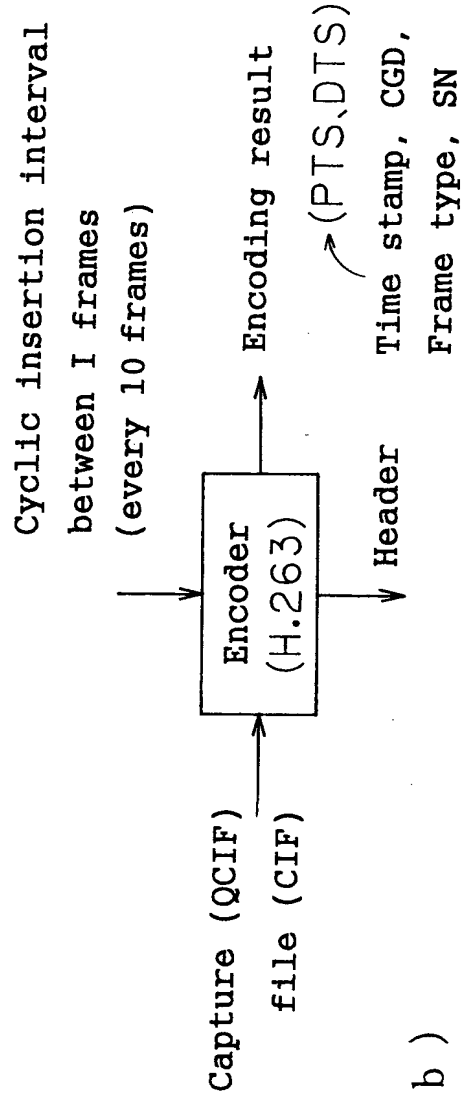


Fig. 20(b)

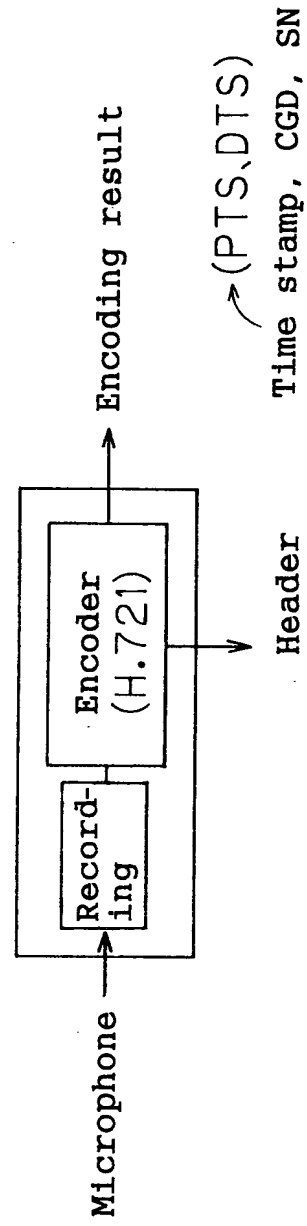


Fig. 20(c)

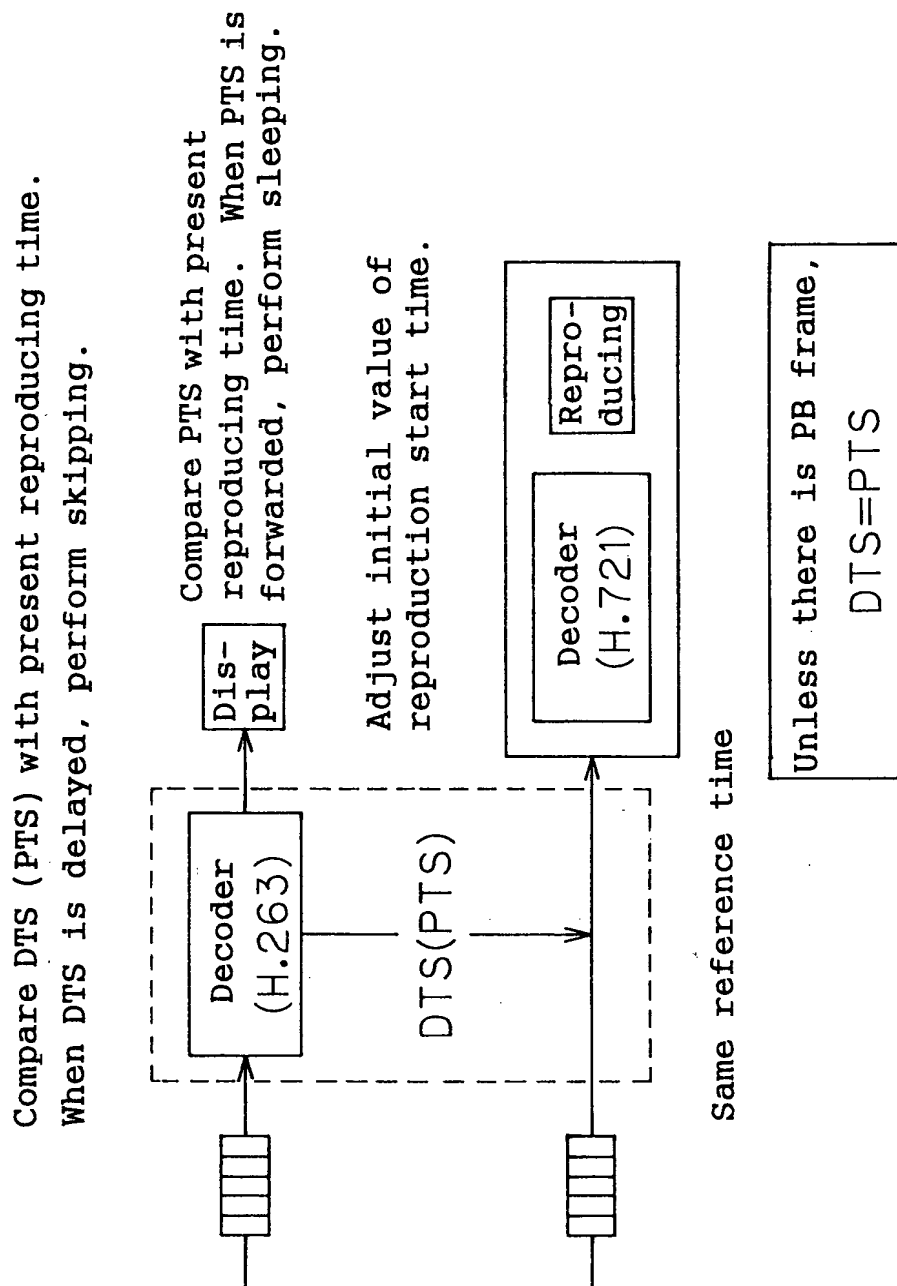
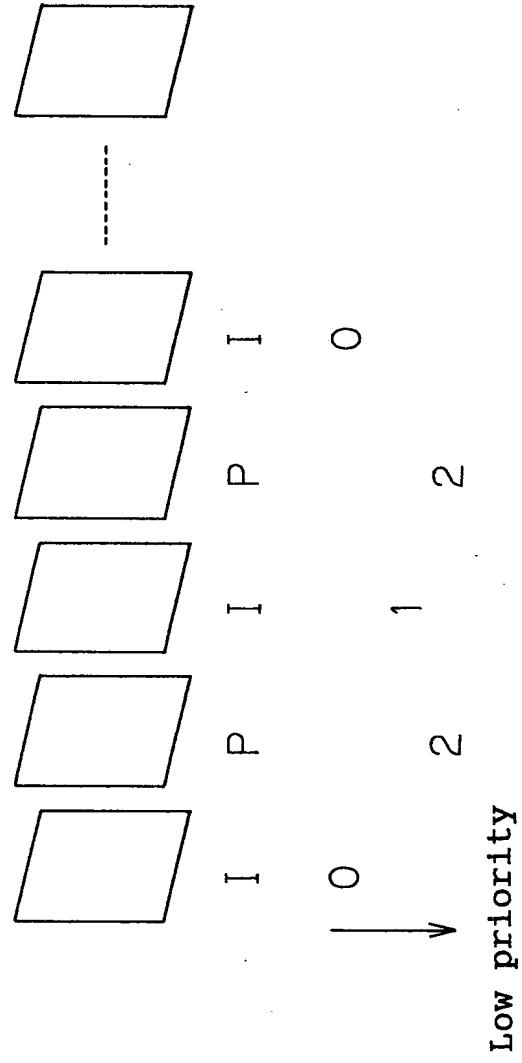


Fig. 21



Equalize priorities of P frames each other and those of PB frames each other.

Processing at receiving terminal under overload (Common to dynamic picture and sound)

Thread for processing sound at system level is previously set its processing priority to a value higher than that of thread for processing picture.

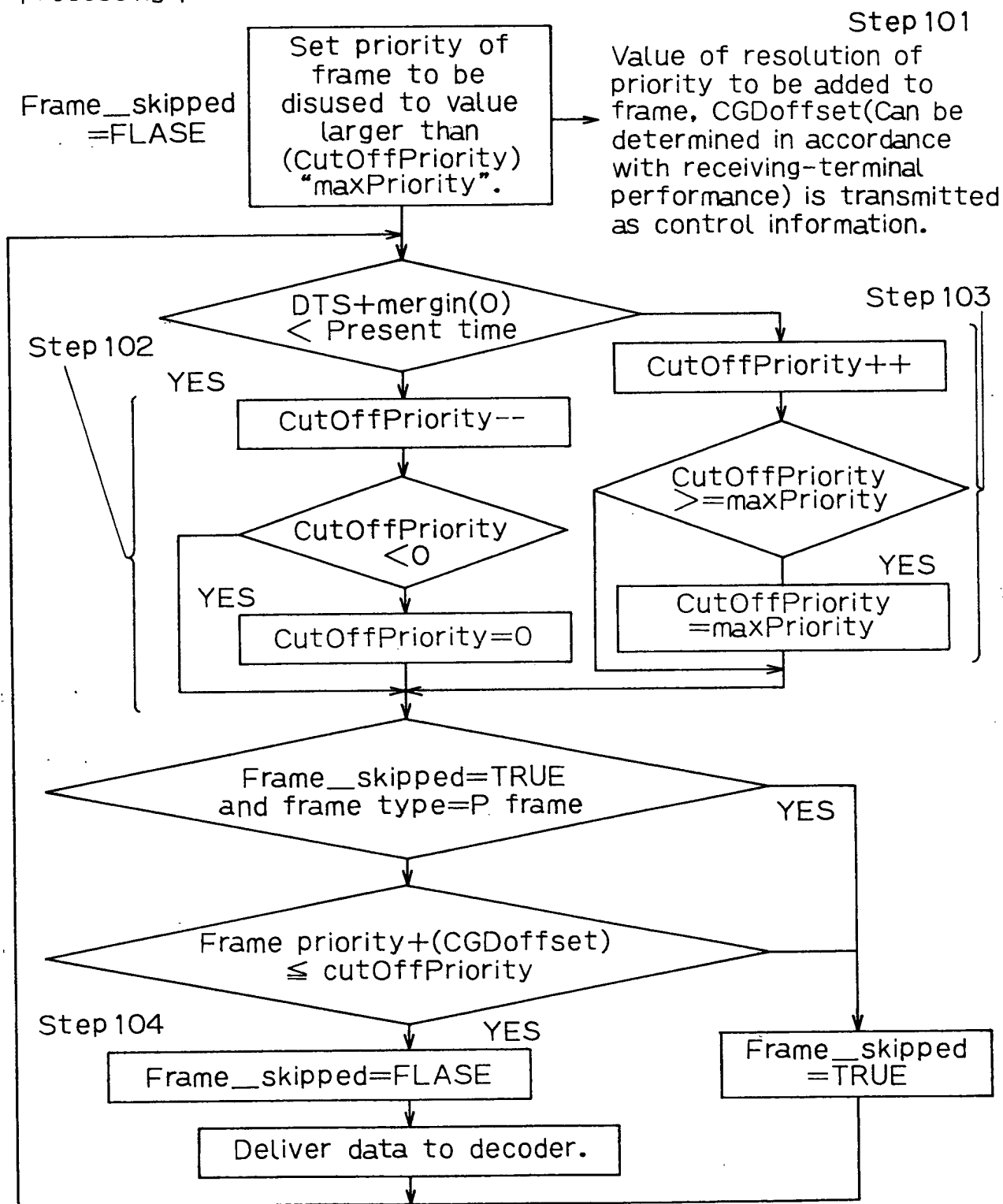


Fig. 23

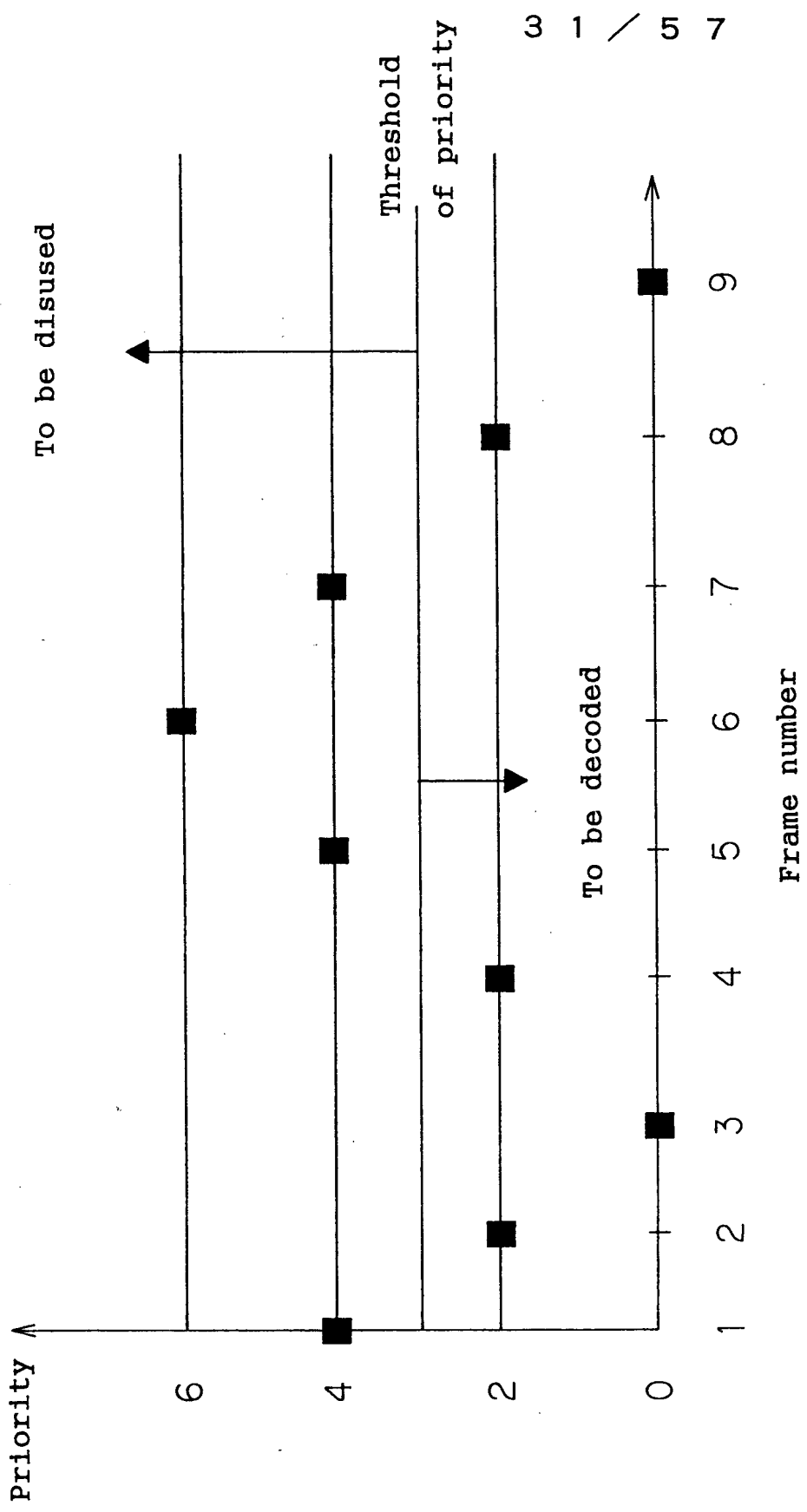


Fig. 24

3 2 / 5 7

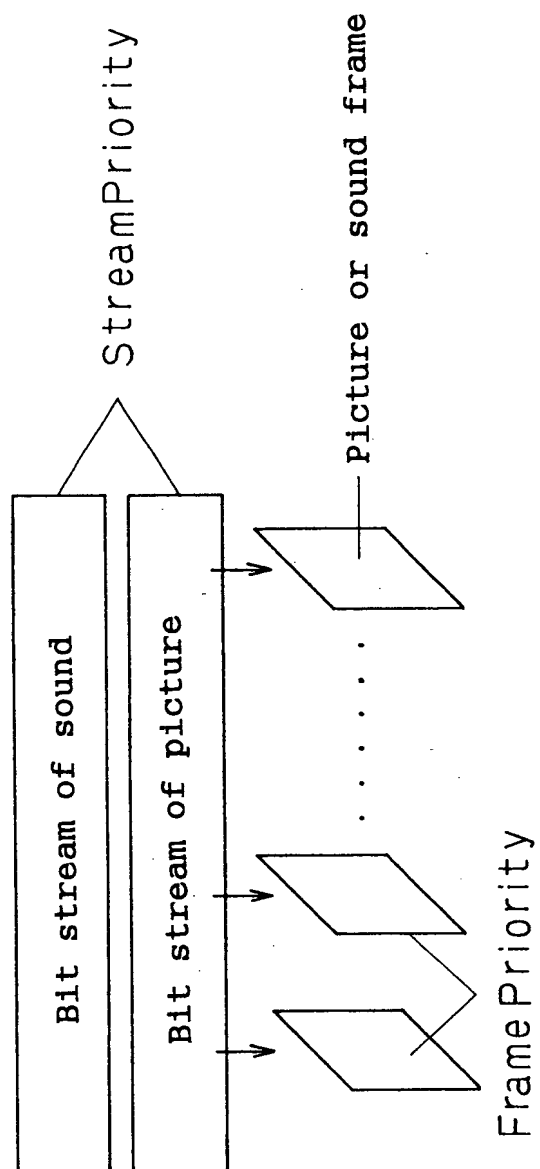


Fig. 25

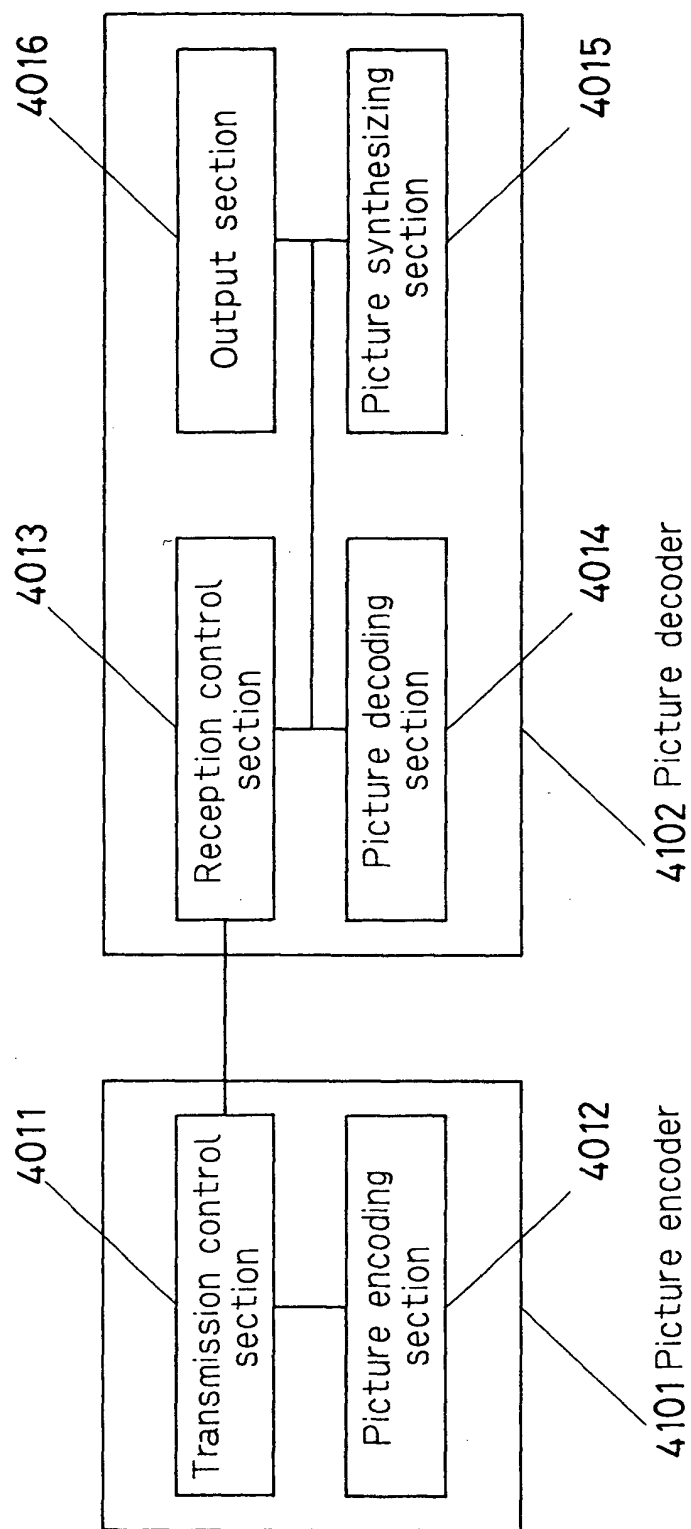
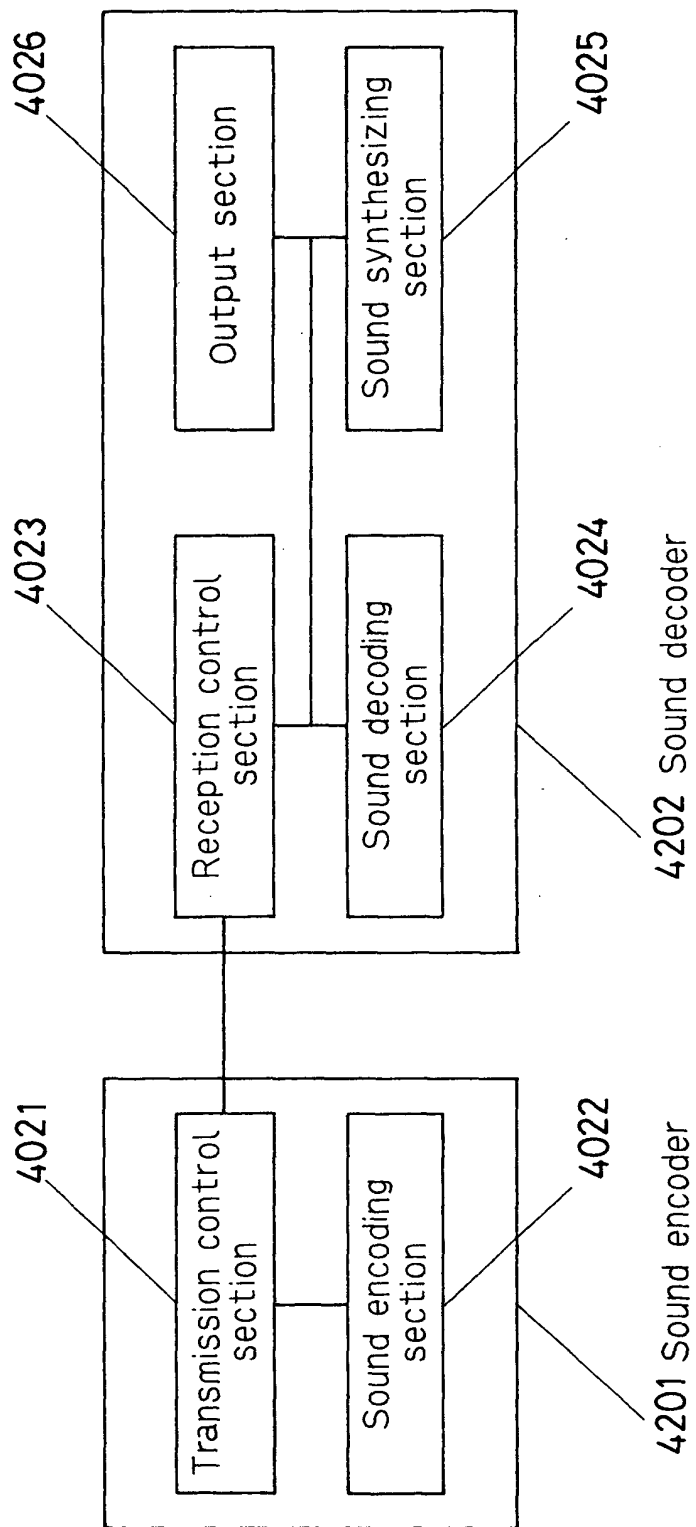
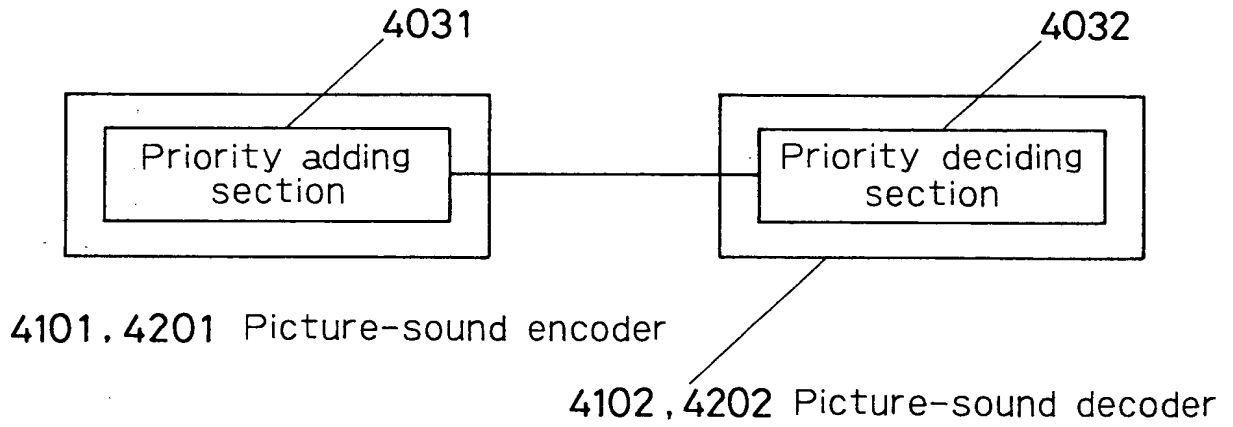


Fig. 26



F i g . 2 7 (a)



F i g . 2 7 (b)

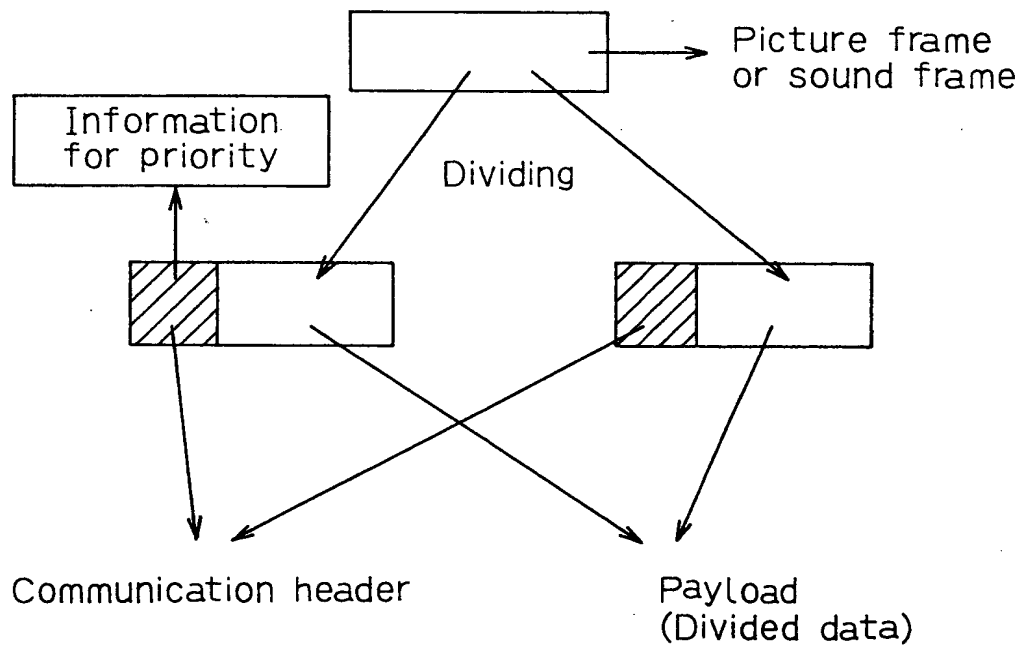


Fig. 28(a)

Relation between StreamPriority and FramePriority

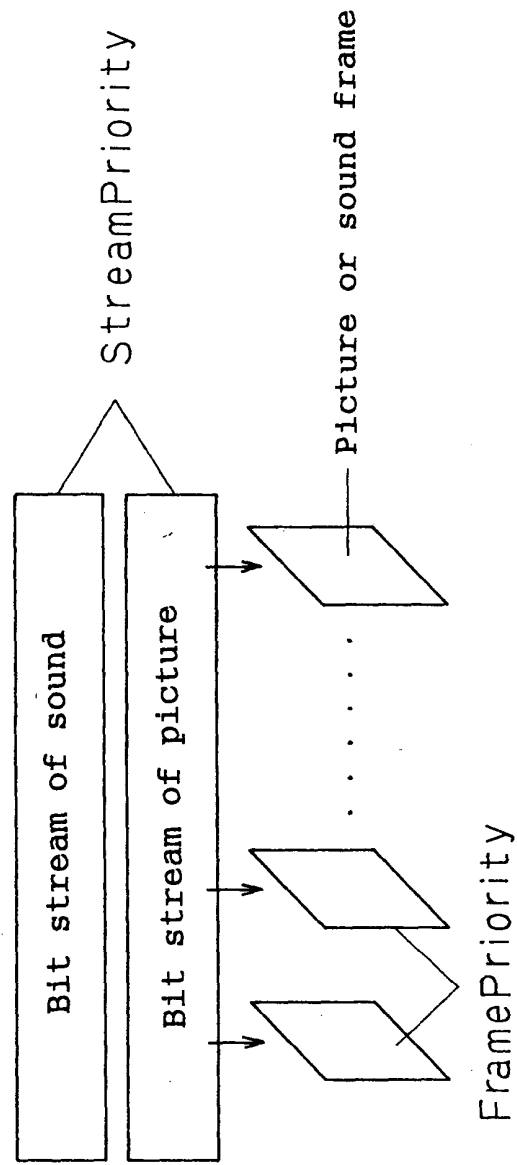


Fig. 28(b)

Priority expressing method (Absolute value/relative value)

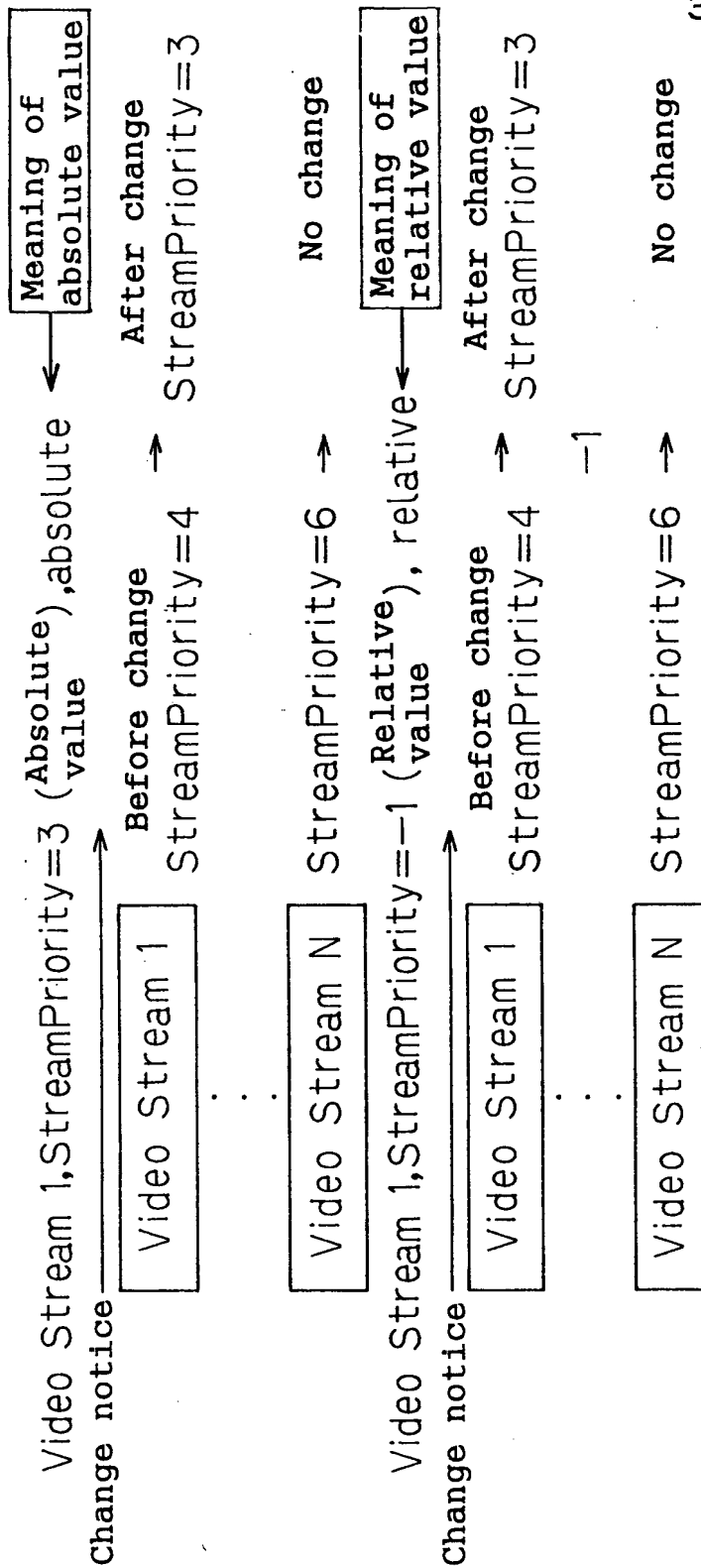
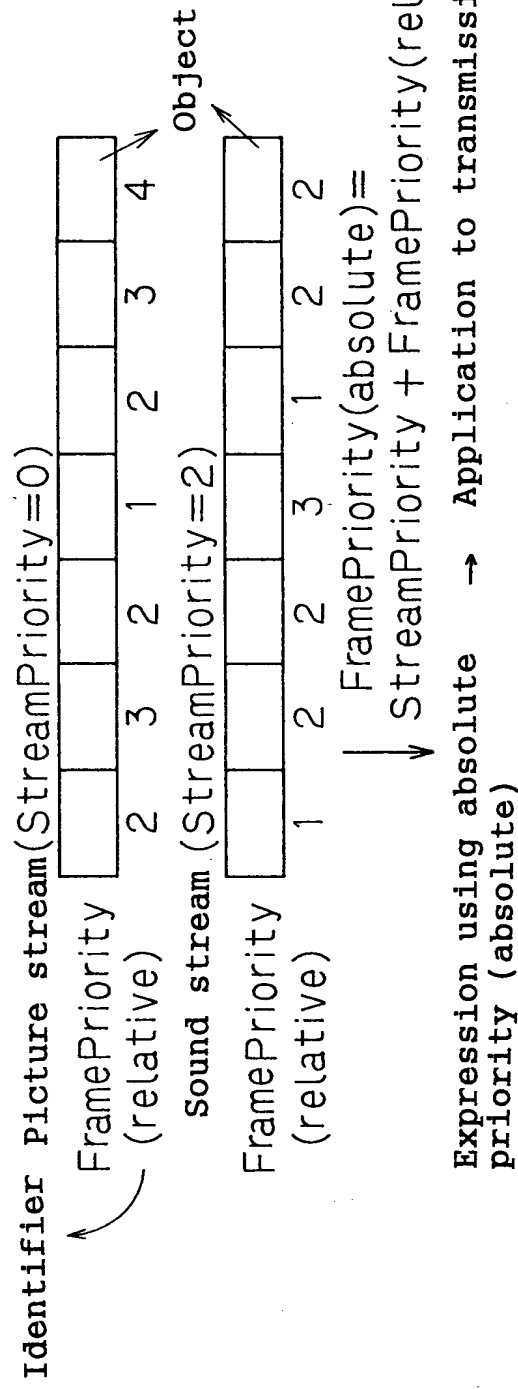


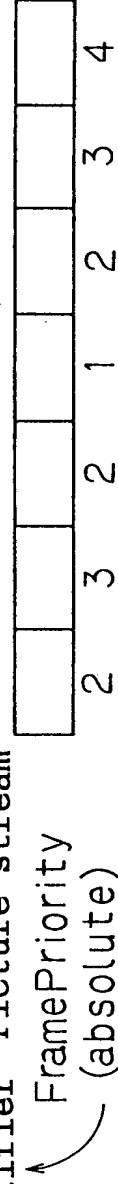
Fig. 28(c)

Expression using relative priority (relative) → Application to accumulation system



3 8 / 5 7

Identifier Picture stream



Sound stream

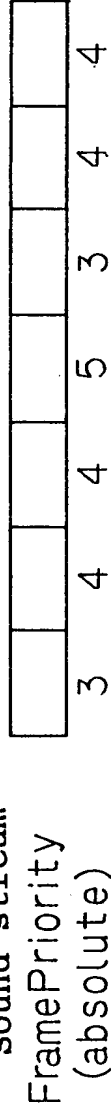


Fig. 29

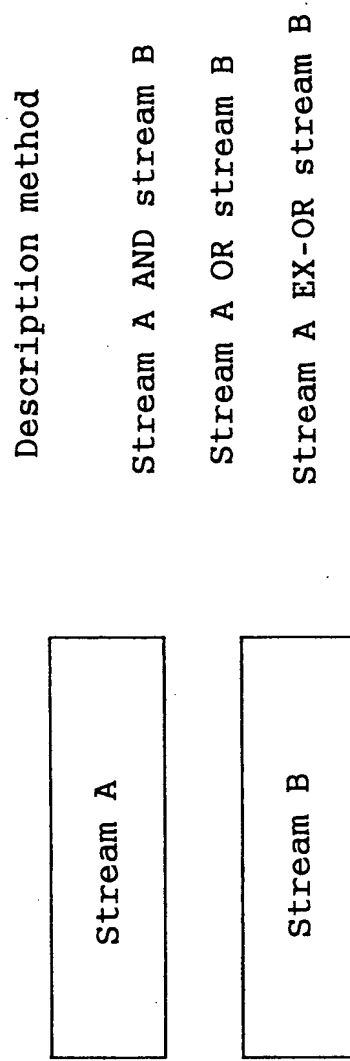
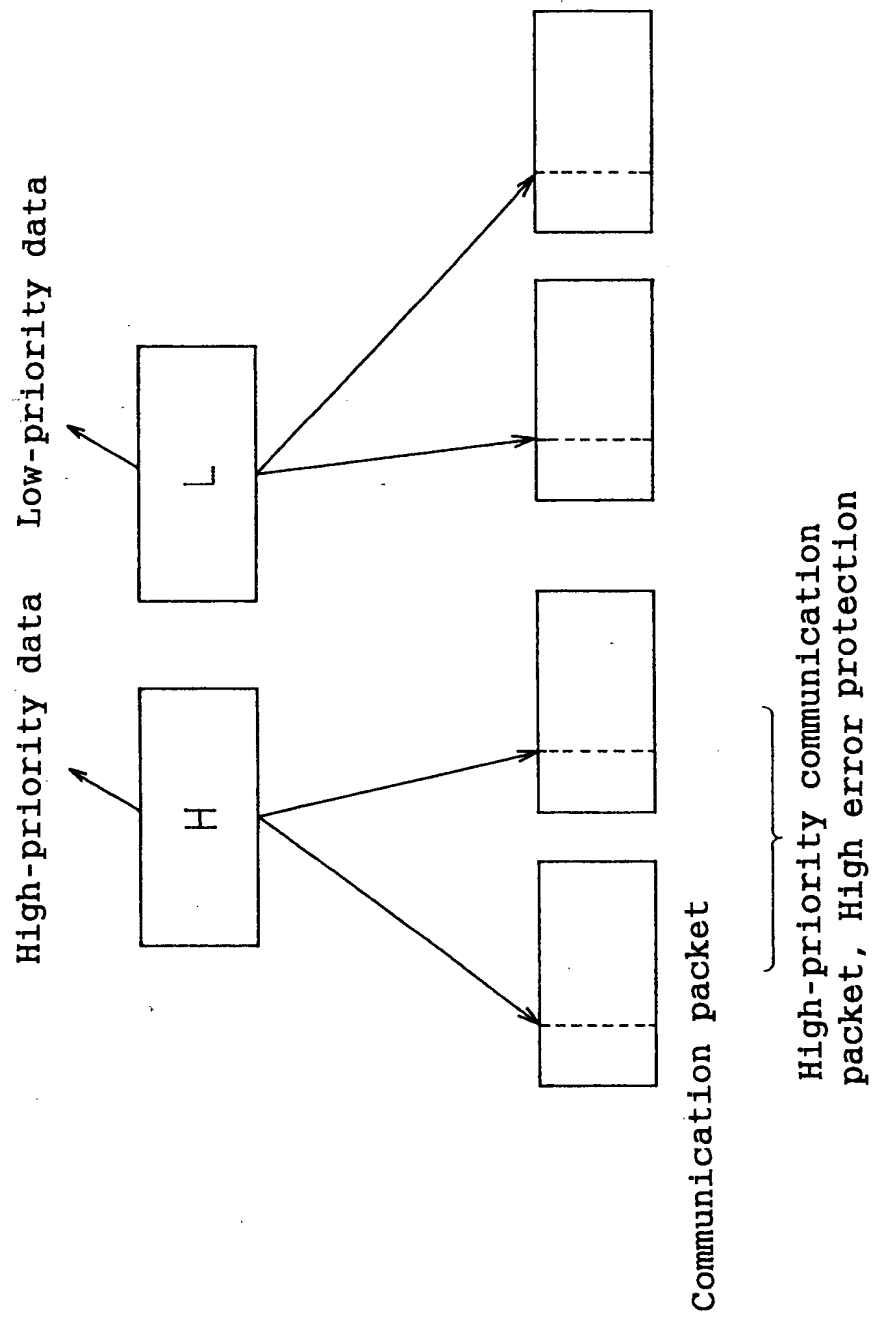
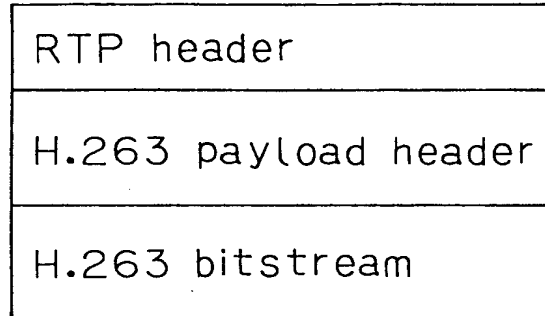


Fig. 30





◦ Mode A: GOB, picture boundary

Presence or absence of mode or PB, start and end positions of bit stream, and execution timing states of options of resolution, frame type, and H.263

→ Core information

DBQUANT, TR(for B frame),
TR(for P frame) → To be set when PB frame is present

◦ Mode B: MB boundary without PB

Core information for Mode A

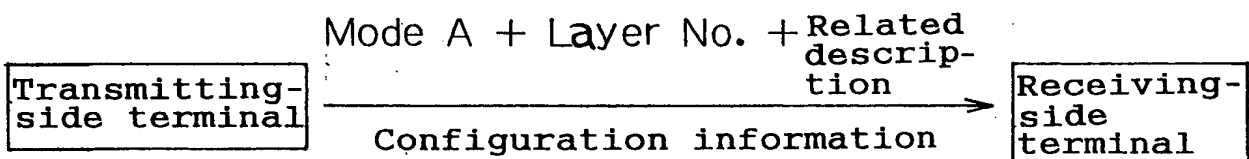
Information for quantization value (GQUANT), GOB number, absolute address of first MB in GOB, and movement vector (Horizontal and vertical directions)

◦ Mode C: MB boundary with PB

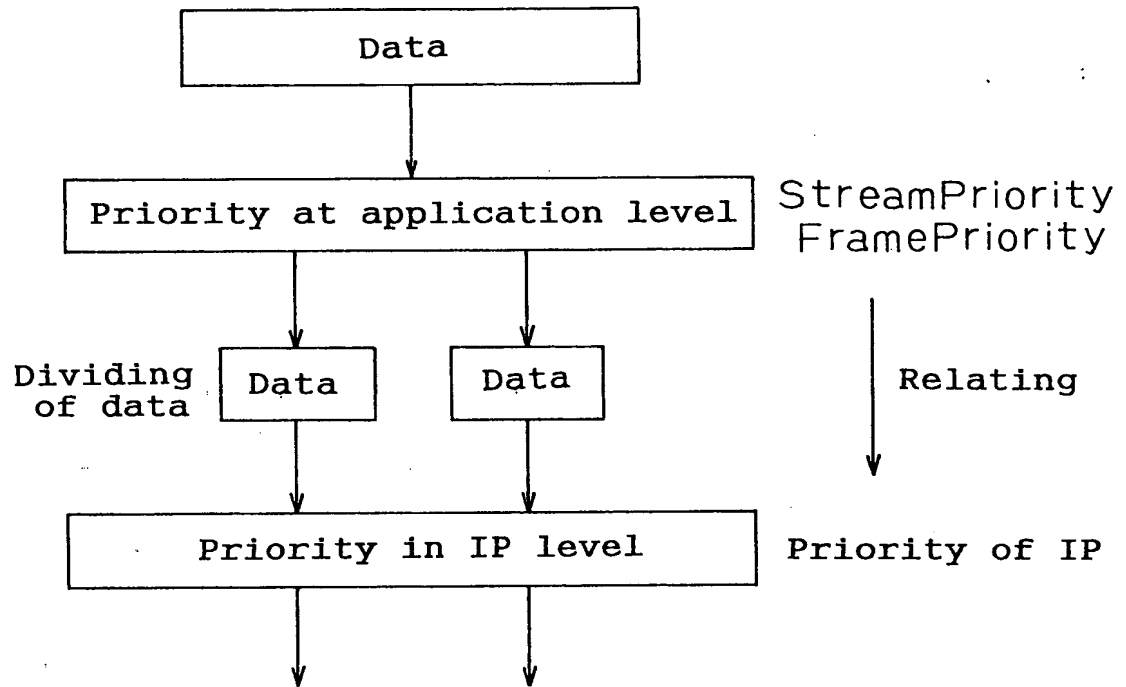
Information for Mode B

DBQUANT, TR(for B frame), TR(for P frame)

Relating of communication payload



F i g . 3 2



Priority in data		Available range	
StreamPriority	0 ~ 3	} [0...15]	Mapping of part
FramePriority	0 ~ 5		
IPV6	8 ~ 15	→ [8...15]	
(Lowest) (Maximum priority)			

F i g . 3 3

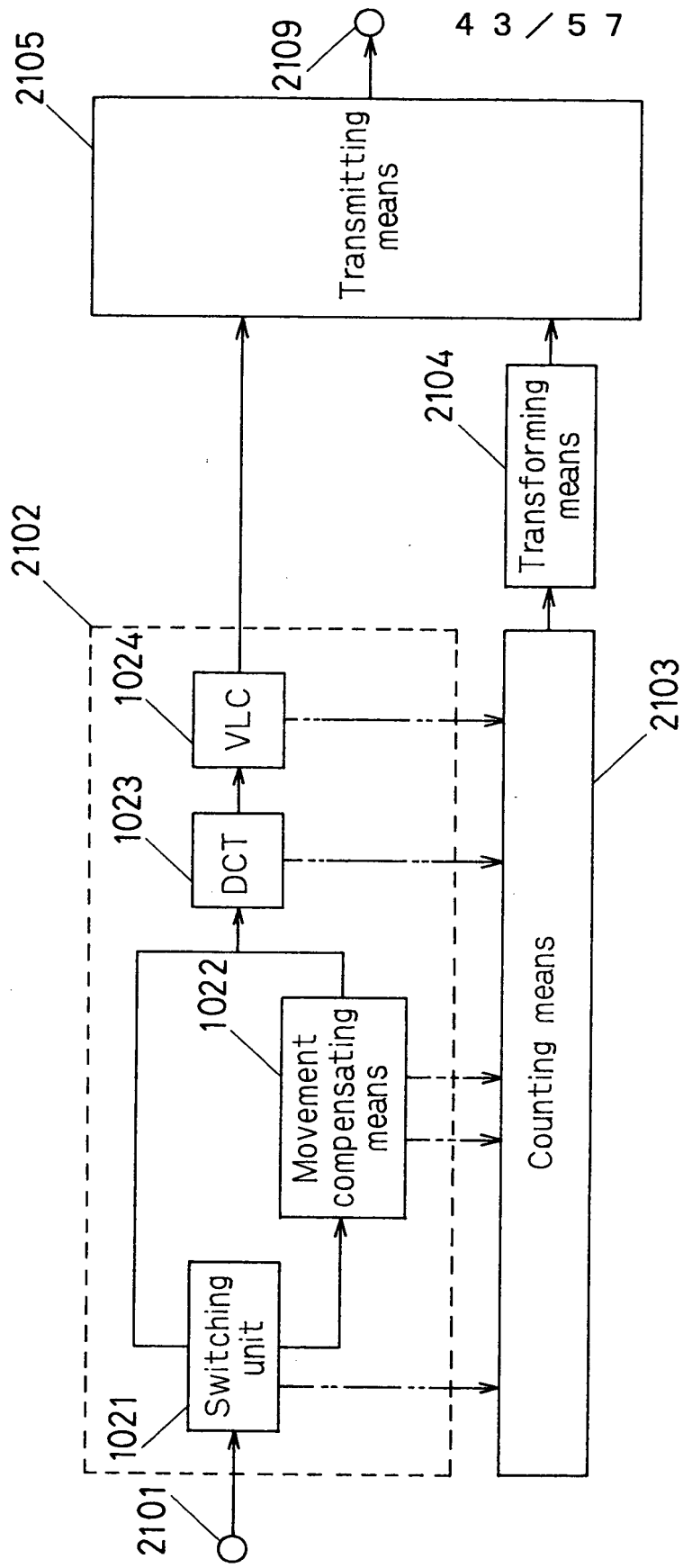


Fig. 34

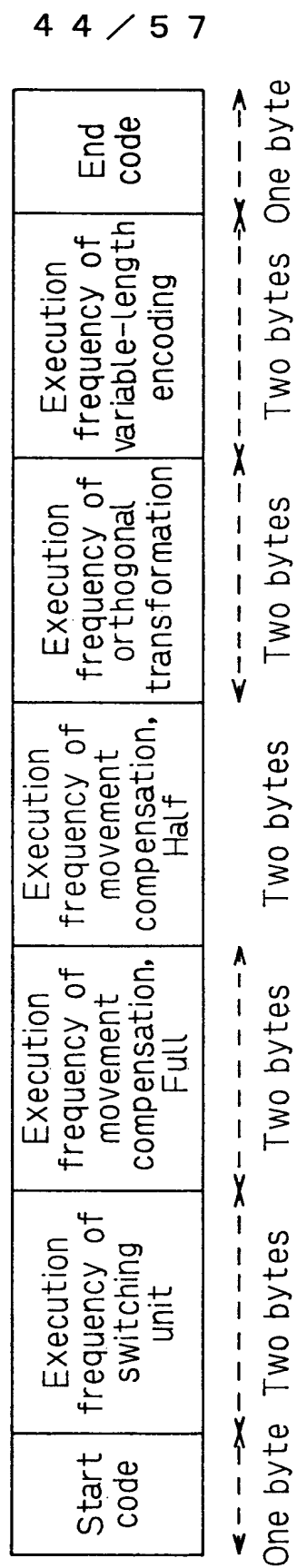


Fig. 35

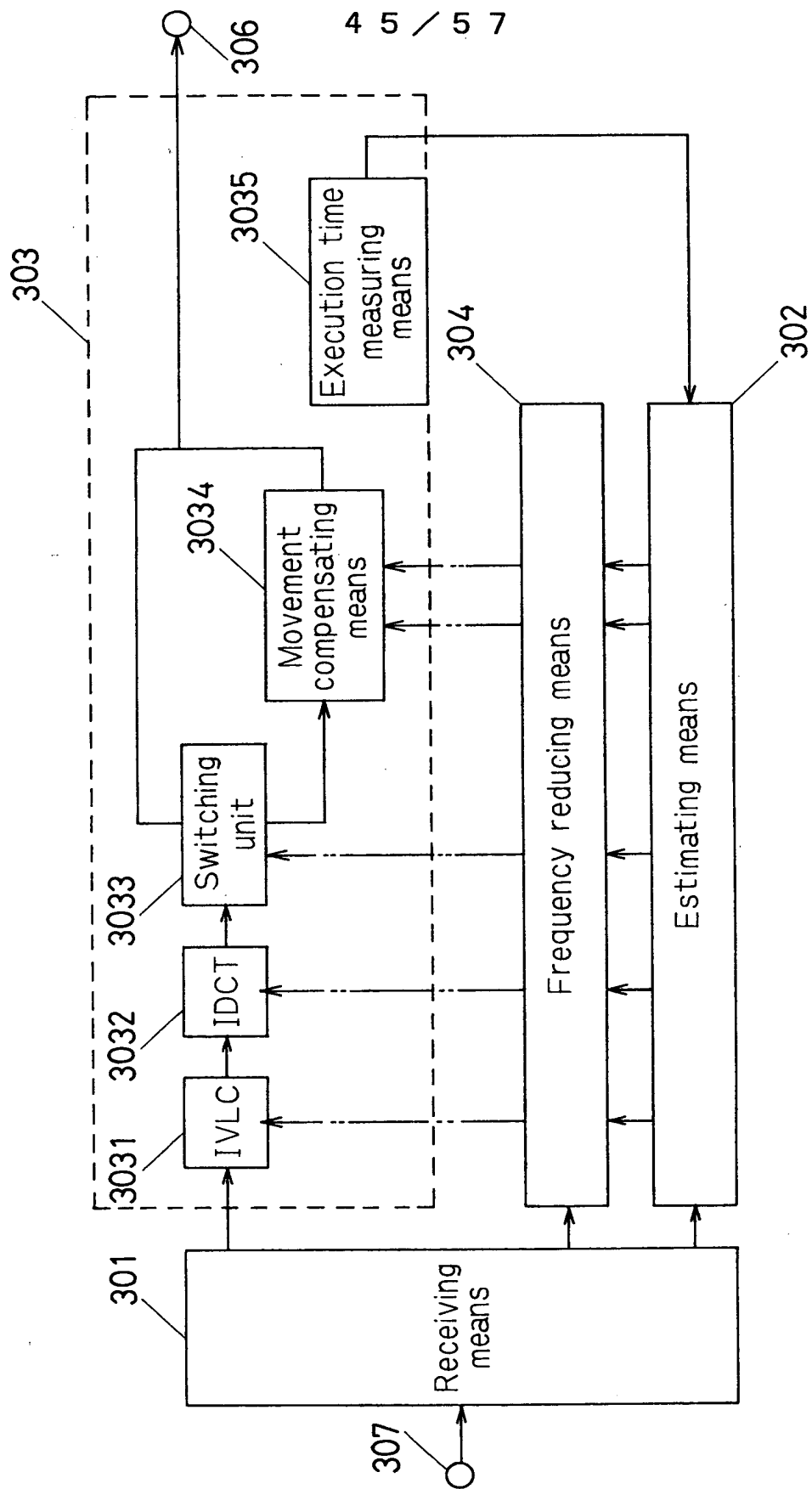


Fig. 36

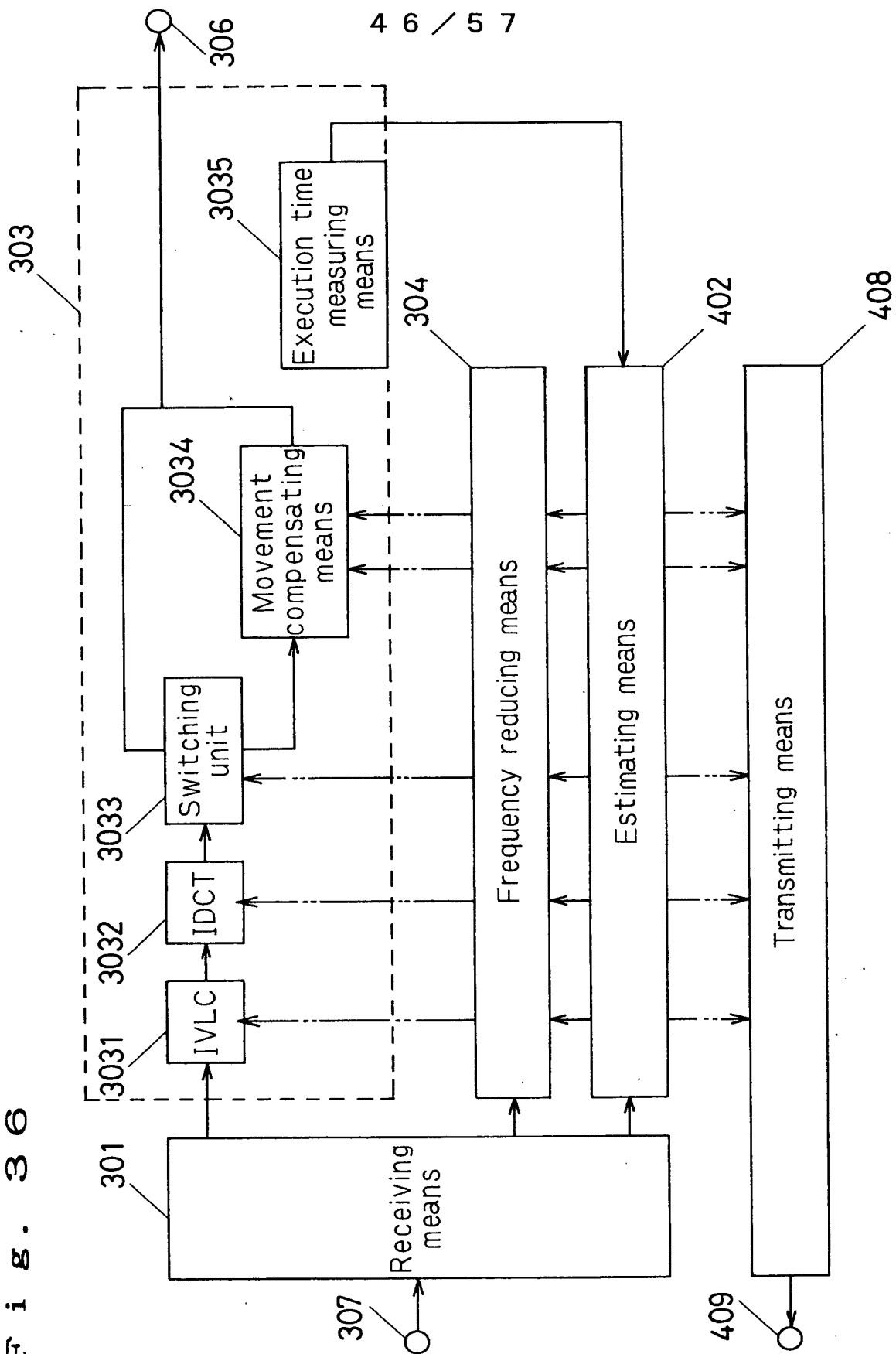


Fig. 37

4 7 / 5 7

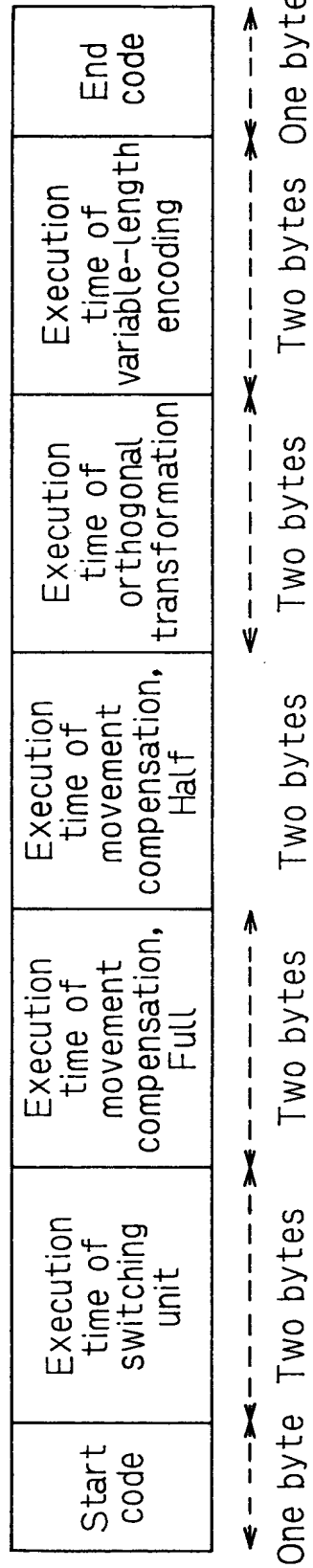


Fig. 38

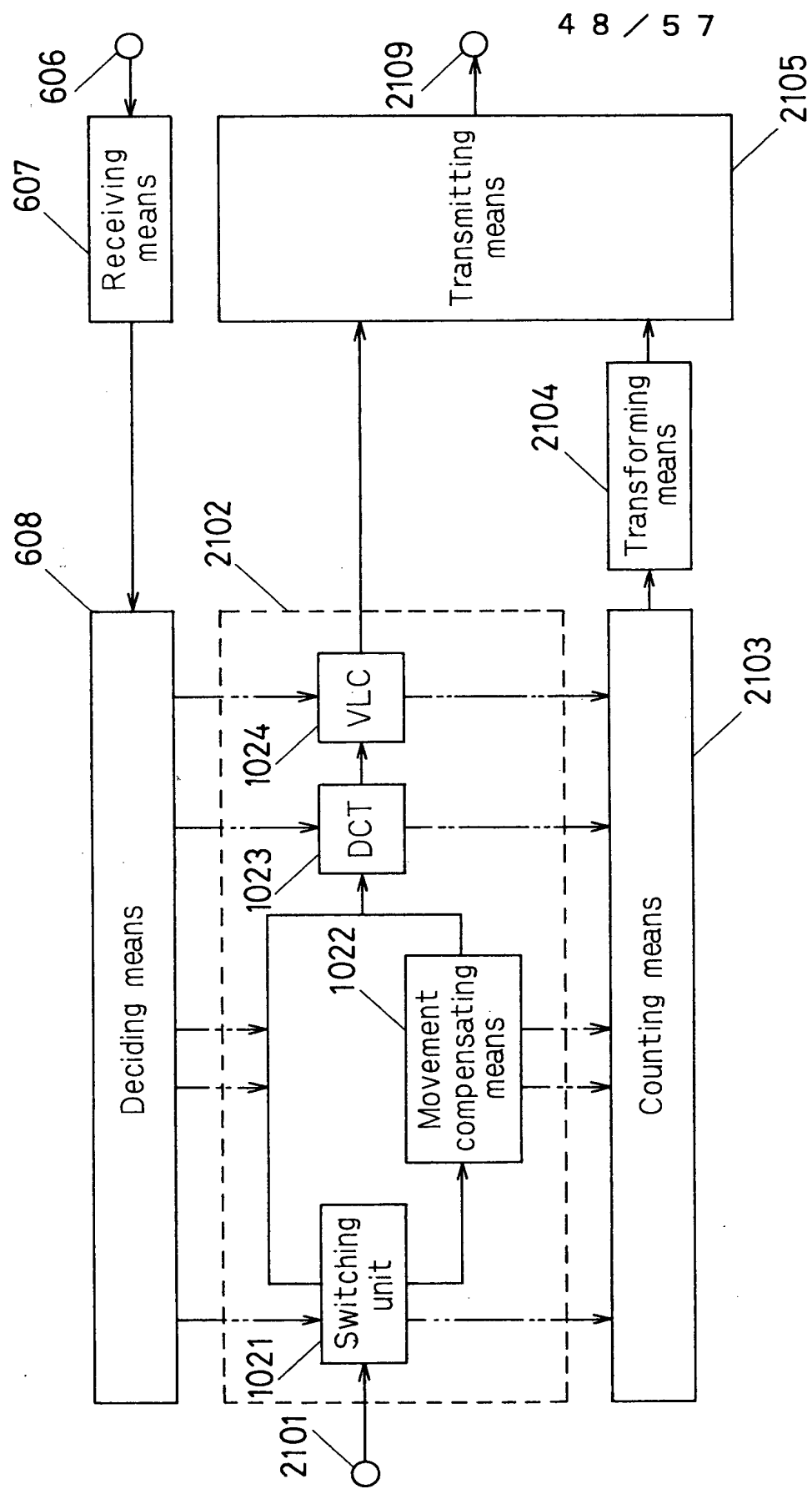
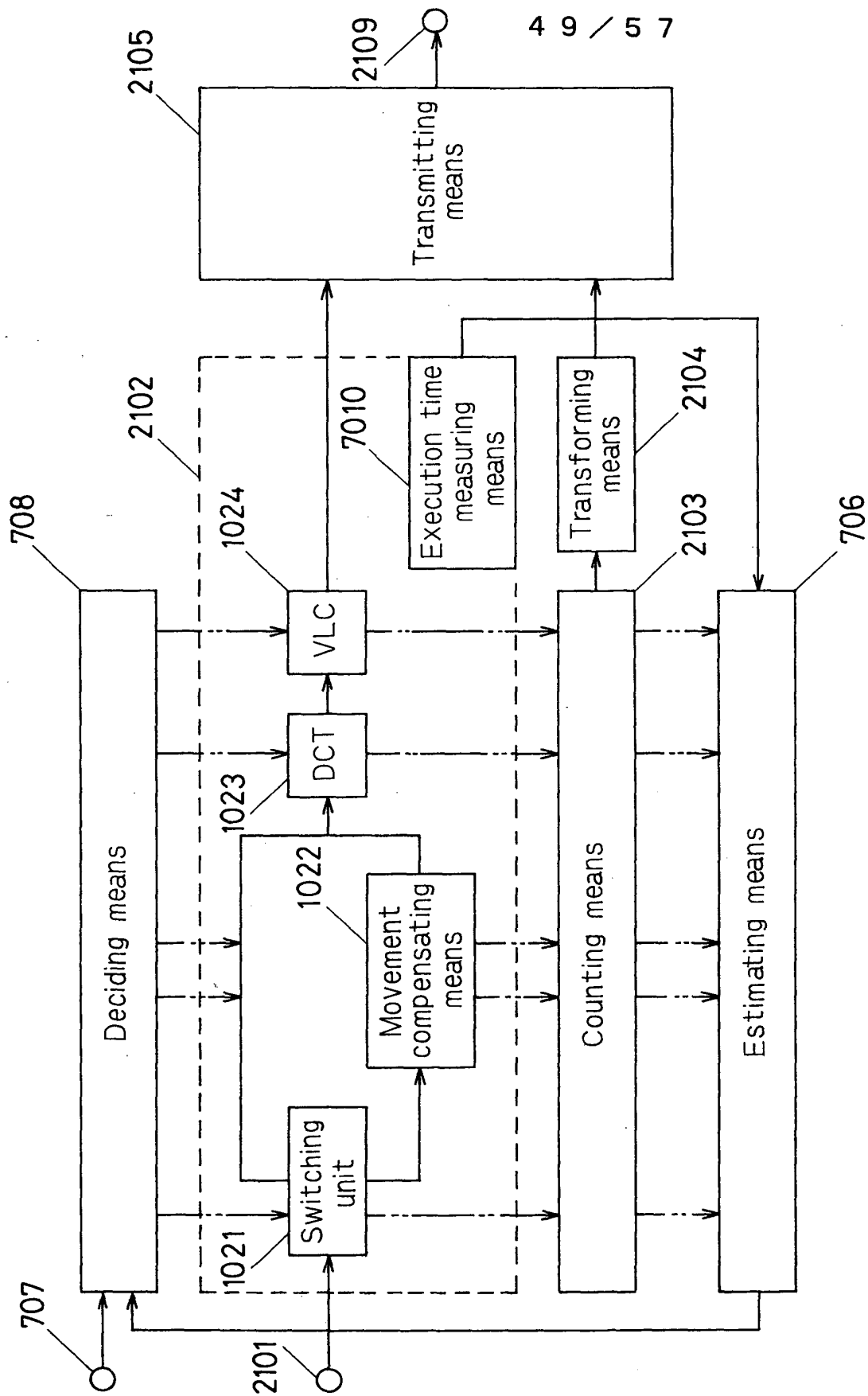


Fig. 39



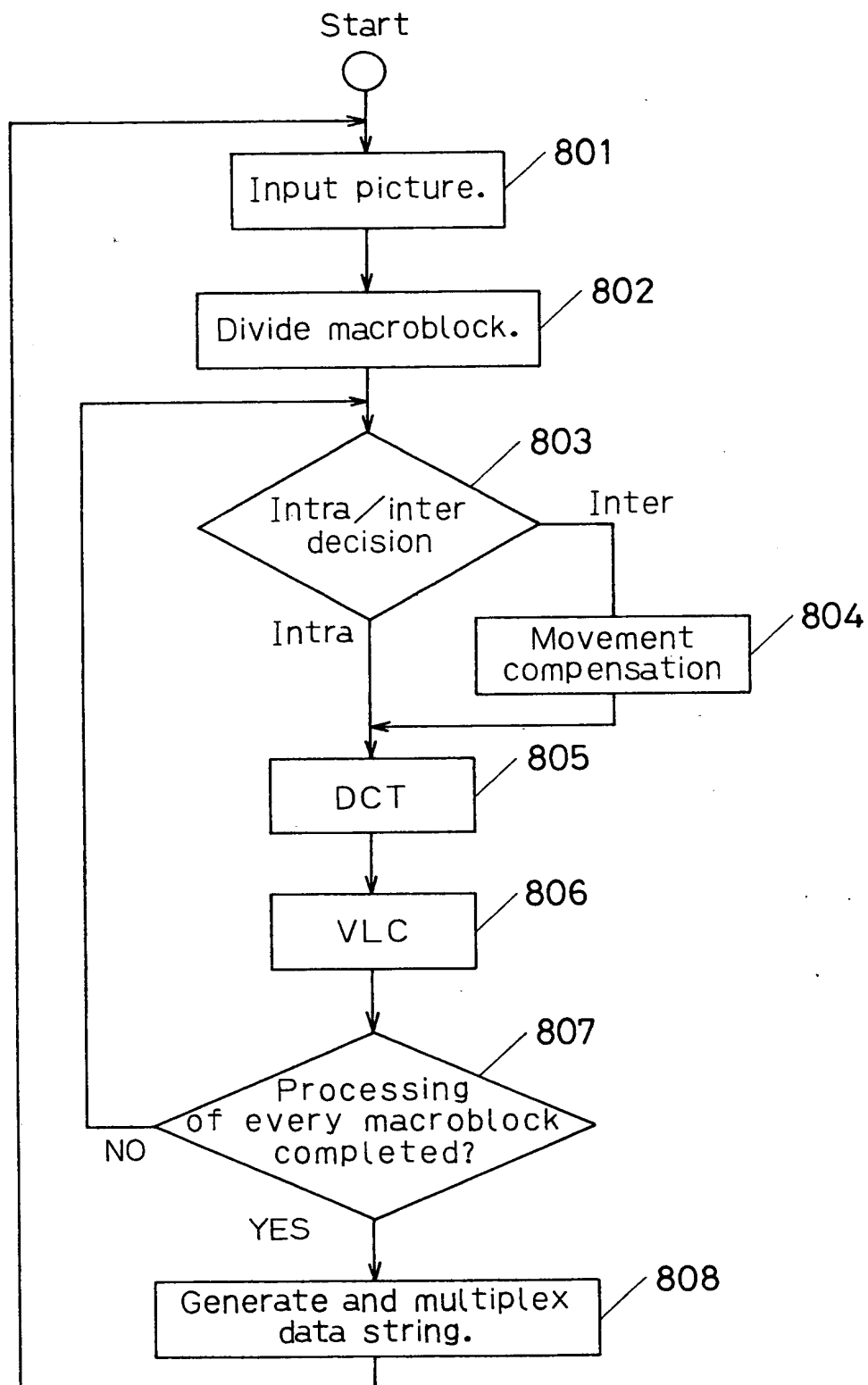


Fig. 41

51/57

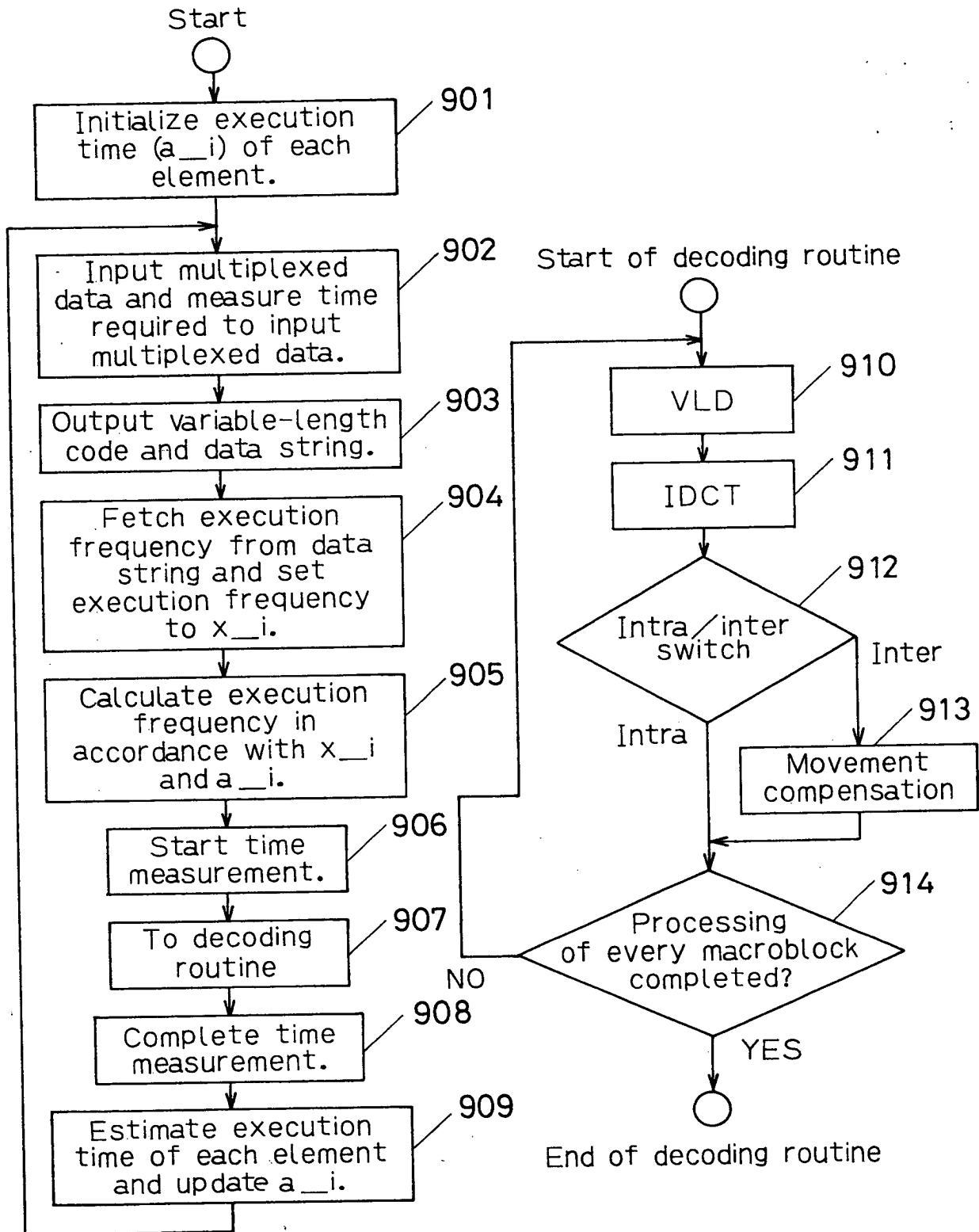


Fig. 42

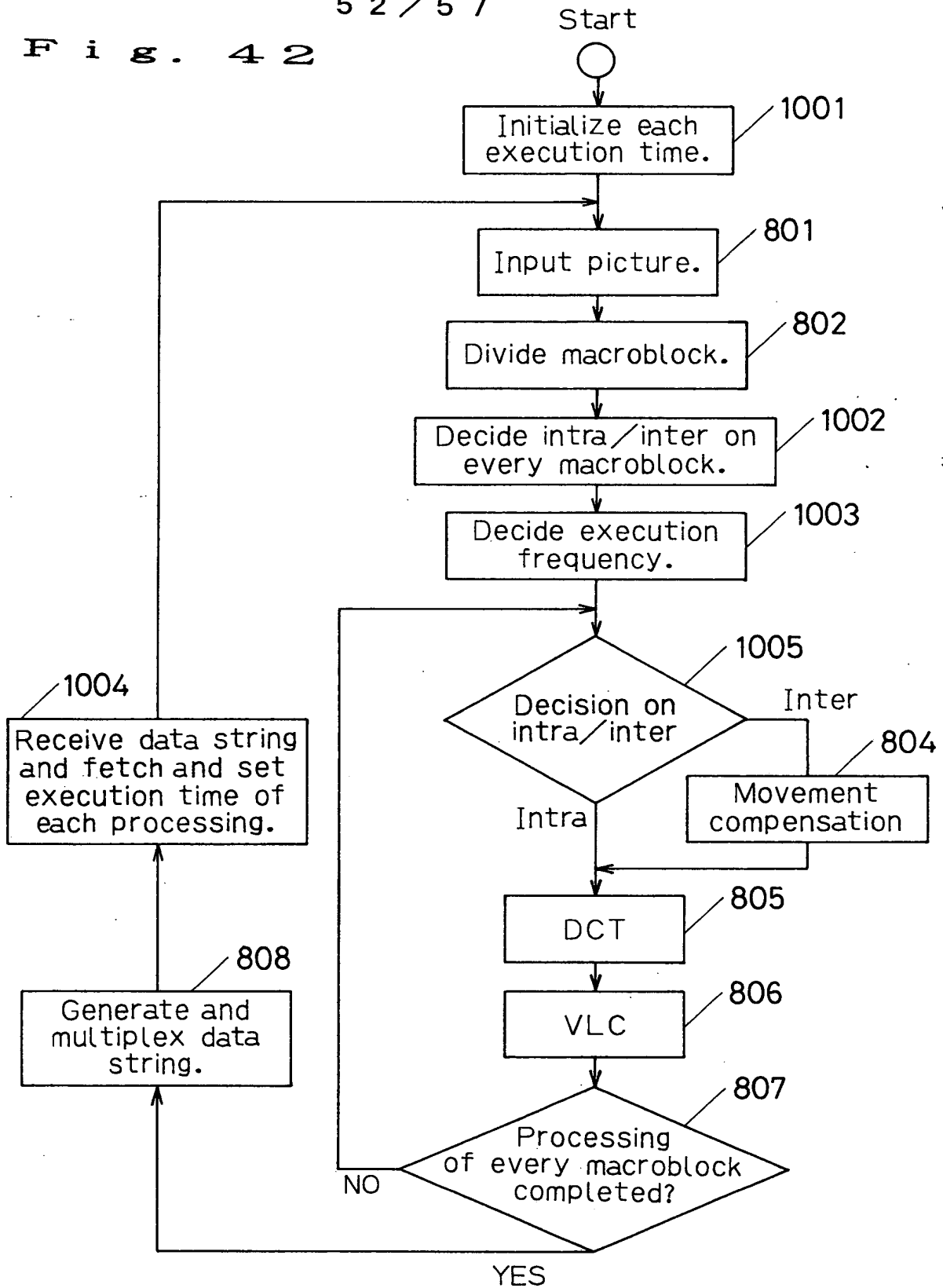
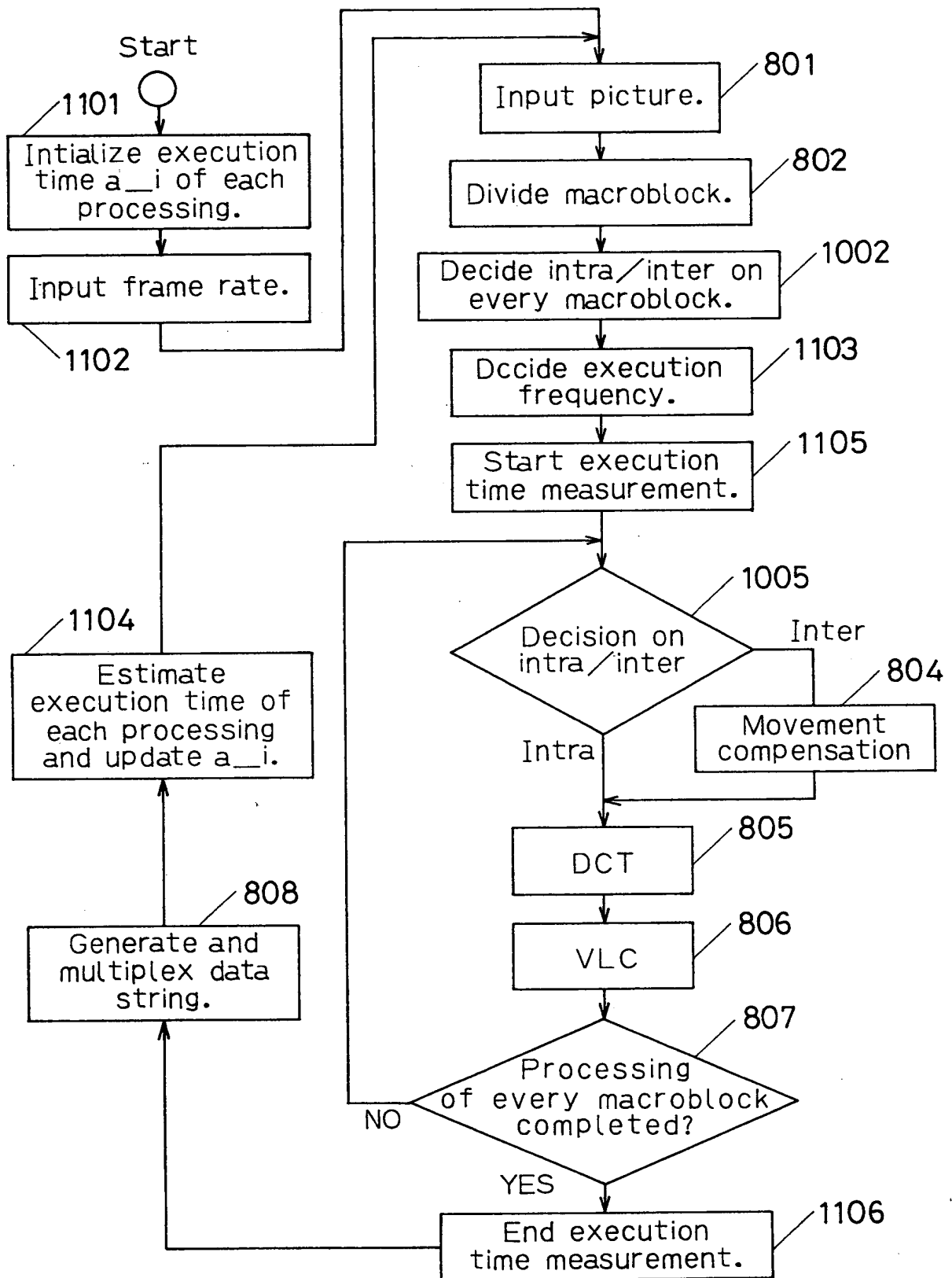


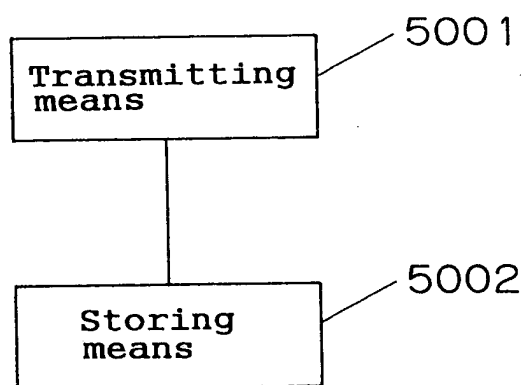
Fig. 43

53 / 57



5 4 / 5 7

F i g . 4 4



F i g . 4 5

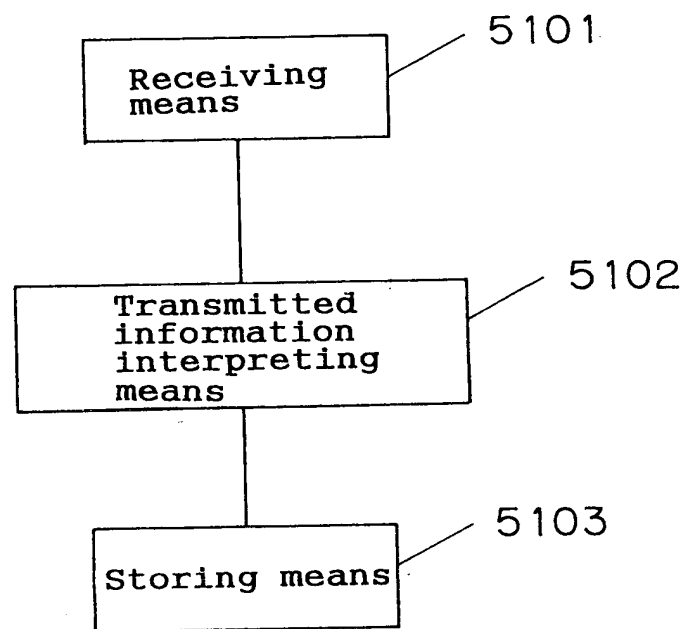


Fig. 46

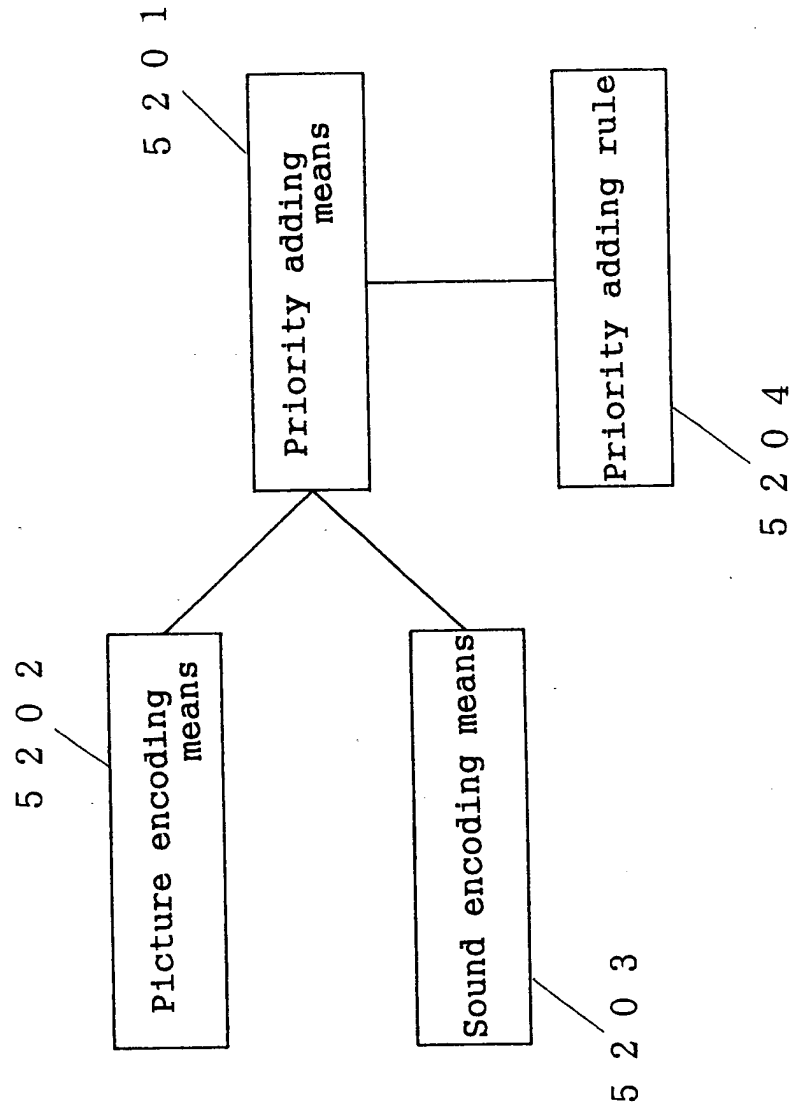
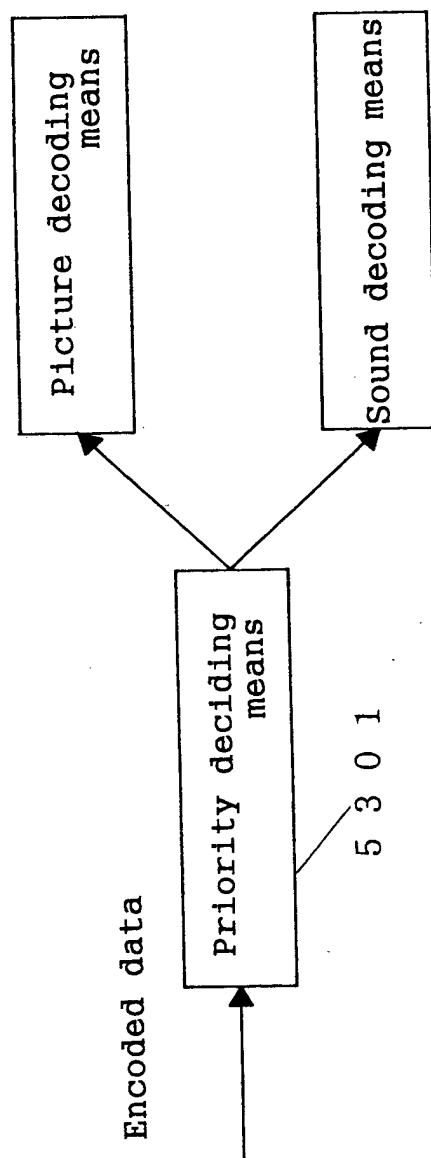


Fig. 47



5 7 / 5 7